

SUPER

n° 4 Ottobre 1984

L. 7.500

Spedizione in
abbonamento
postale Gruppo III/70

4

SUPERVIC
la rivista per gli utenti
dei prodotti Commodore

VIC & C64

Una pubblicazione della

J. soft EDITRICE

in collaborazione con

GRUPPO
EDITORIALE
JACKSON



LO SCALATORE FOLLE

**GRAFICA BITMAP
PER C64**

DISEGNATORE

TERAPIA

TRIDENT

MOZART



Soft Dealer

La rivista per chi vende personal computer - software - libri - accessori

Quali sono i pacchetti e i libri di informatica più venduti in Italia?
È proprio vero che tutti vogliono un home computer?
Lotus 1-2-3 è solo una moda?
Apple II C è veramente compatibile con II E?
Chi copia il software?
Come fare business con il software?

A QUESTE
E MOLTE ALTRE
DOMANDE RISPONDE **Soft Dealer**

Il primo mensile italiano per chi vende home e personal computer, software, libri e accessori di informatica. Uno strumento utilissimo per chiunque si interessi di personal e home computer. Su **Soft Dealer**, il nuovo mensile della PROGRAM EDITRICE, inviato solo in abbonamento, troverete tutte le notizie più utili, recensioni, test, prove di software, articoli di marketing, classifiche dei migliori programmi esistenti sul mercato.

UN ABBONAMENTO A **Soft Dealer**
È UN INVESTIMENTO PER IL FUTURO. APPROPFITTA DELL'ECCEZIONALE OFFERTA DI LANCIO.
RICEVERAI GRATIS I PRIMI TRE NUMERI DEL 1984.

Per abbonarsi basterà compilare il modulo pubblicato qui sotto, allegando assegno di L. 25.000, e inviarlo a PROGRAM EDITRICE S.r.l. Via Melchiorre Gioia, 114 - 20125 Milano.

SOFT DEALER è un'idea

PROGRAM
editrice

Uffici Tecnici e Commerciali:
20125 Milano - Via M. Gioia, 114
Tel. (02) 603415-680851

- ☐ Desidero sottoscrivere un abbonamento a 10 numeri (L. 25.000) alla rivista **SOFT DEALER**, da gennaio a dicembre 1985.
In questo caso mi saranno inviati in omaggio i primi tre numeri del 1984.
- ☐ Allego assegno di L. 25.000 intestato a PROGRAM EDITRICE
n° Banca
- ☐ Ho effettuato pagamento tramite vaglia postale.

NOME

COGNOME

AZIENDA

VIA N°

C.A.P. CITTA'

SUPER

VIC & C64

5 EDITORIALE POINT

PIRATEGGIANDO
di Pietro Dell'Orco

7 &WRITE LA POSTA DEI LETTORI

9 OPEN

**COMMODORE 64
COMMODORE PLUS/4
HOME BABYSITTER**
a cura della Redazione

13 ...then

**INIZIARE CON UN
SISTEMA A DISCHI**
parte seconda: primi passi
di C. Brannon
trad. e adatt. di
M. Cristulib Grizzi

17 REM:HW

**TUTTO QUELLO CHE
AVRETE VOLTATO SAPERE
SUL REGISTRATORE A
CASSETTE...**
di M. Cristulib Grizzi

24 LOAD

**LO SCALATORE
FOLLE**
di T. Reynolds
trad. e adatt. di
G. G. Barzaghi

**LA MEMORIA
NASCONDA DEL C64**
di J. R. Krauss
trad. e adatt. di
M. Cristulib Grizzi

**UNA FINESTRA SUL
VIC 20**
di G. Dan
trad. e adatt. di
R. Comini

**GRAFICA "BITMAP"
PER IL C64**
di M. Tingle
trad. e adatt. di
G. G. Barzaghi

TERAPIA
di S. Rubio
trad. e adatt. di
P. Ferrari

INGRANDIRE PER C64
di V. Aglio

DISEGNATORE
di D. E. Smith
trad. e adatt. di
G. G. Barzaghi

58 TRENCHFIRE
di D. Gibson
trad. e adatt. di
M. Cristulib Grizzi
e F. Sarcina

61 TRIDENT PER C64
di C. O. Dickerson
trad. e adatt. di
F. Sarcina

68 VICsound

NOISE MAKER PER C64
di P. Ferrari
MOZART
di D. J. Eddington
trad. e adatt. di
E. Comini e M. Cristulib Grizzi

76 RUN

RECENSIONI SOFTWARE
di M. Cristulib Grizzi

78 PRINT

RECENSIONI LIBRI
di M. Cristulib Grizzi

80 input-output

PICCOLI ANNUNCI

J. soft s.r.l.
**DIREZIONE, REDAZIONE,
AMMINISTRAZIONE**
Via Rosellini, 12
20124 MILANO
Tel. (02) 68.88.228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paoletti

REDAZIONE:
Mauro Cristulib Grizzi

**HANNO COLLABORATO A
QUESTO NUMERO**
G. Barzaghi
R. Comini
P. Ferrari
V. Aglio
F. Sarcina
E. Comini

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Raffaella Toffolatti

FOTOCOPOSIZIONE:
d&h Via Vignola, 5
Tel. 02-56.85.08
20133 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonati

**AUTORIZZAZIONE ALLA
PUBBLICAZIONE:**
Tribunale di Milano n° 201
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionario per l'Italia e
l'Estero Reina s.r.l.
Via Washington, 50
20046 MILANO
Tel. (02) 49.88.066 (5 linee R.A.)
Tlx. 316213 REINA I

Concessionario esclusivo per la
DISTRIBUZIONE in Italia e Estero
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento
postale Gruppo 111/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo (11 numeri)
L. 75.000; per l'estero L. 110.000
I versamenti devono essere
indirizzati a:
Jsoft s.r.l.
Via Rosellini, 12
20124 MILANO

mediante emissione di assegno
bancario, cartolina, vaglia o
utilizzando il c/c postale
n. 19445204
Per i cambi di indirizzo indicare,
oltre al nuovo, anche l'indirizzo
precedente ed allegare alla
comunicazione l'importo di L. 500,
anche in francobolli

© TUTTI I DIRITTI DI
RIPRODUZIONE O TRADUZIONE
DEGLI ARTICOLI PUBBLICATI
SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO
TRATTI DALLA RIVISTA COMPUTE!
O DA COMPUTE! GAZETTE.
COMPUTER PUBLICATION, INC.
PO BOX 5426
GREENSBORO, NC 27403 - USA

Sul prossimo numero troverete:



SPEEDSCRIPT: Un vero word processor di qualità commerciale per il vostro VIC 20 o C64, interamente in linguaggio macchina ed all'altezza dei migliori prodotti esistenti sul mercato.



ULTRAFONT: Un generatore di caratteri per C64, totalmente in linguaggio macchina, con possibilità di creare caratteri multicolori, immagini a specchio, ruotate, invertite... Il massimo nel suo genere!



SPIRALIZER: Con questo programma potrete creare immagini fantastiche in alta risoluzione sullo schermo del vostro computer.



TANTI ALTRI giochi e programmi di utilità per i vostri VIC 20 e C64 assetati di... **SUPER!**

IN MILANO - VIA MASCHERONI, 14

IL VOSTRO "NUOVISSIMO" COMPUTER SHOP

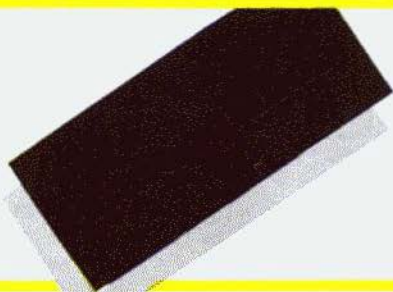
Libri e riviste di elettronica e informatica.

**CORSI CONTINUI
TUTTO L'ANNO
CON I MIGLIORI
SOFTERISTI**



IL TELEFONO È 02-437.385

POINT



Sempre con maggiore frequenza si assiste al "fenomeno", diciamo così, della pirateria del software. Non che la cosa sia nuova, vale comunque la pena di spendere qualche parola a tal proposito.

Per scrivere del buon software giocoso, professionale o gestionale servono molte ore di duro lavoro intellettuale, un certo numero di ore di lavoro manuale relative alla codifica, oltre al tempo necessario per il test e la correzione del programma.

Forse qualcuno di voi si è cimentato nella scrittura di programmi semplici o complessi e si sarà reso conto di quanta fatica sia necessaria per la messa a punto di un prodotto software.

Data la premessa è chiaro che pirateggiare, rubando il software altrui per venderlo sottocosto è azione degna perlomeno di biasimo se non di azioni legali vere e proprie.

La legislazione italiana non prevede ancora forme di protezione efficaci per chi cerca di vivere ed operare onestamente nel settore del software; di conseguenza individui privi di scrupoli e di etica professionale possono "rubare" impunemente il prodotto altrui frustrando così ogni possibilità di onesti guadagni.

Un vasto numero di operatori del settore sta organizzando iniziative per reprimere questo dilagante malcostume. Per ogni forma di protezione del software messa a punto dagli autori vengono prodotti altrettanti sistemi di copiatura abusiva.

Una situazione senza fine perpetrata in continuazione a scapito di una qualificata offerta di validi prodotti.

Tutto ciò crea la necessità di elevare i prezzi per recuperare, almeno in parte, l'investimento "tempo" speso per l'elaborazione del programma. I maggiori costi gravano, assurdamamente, sugli utenti onesti che acquistano regolarmente il programma.

Una cassetta con un buon gioco per VIC, C64 o Spectrum potrebbe costare dalle 3 alle 5.000 lire se questo mercato seguisse le normali regole del commercio.

La prossima volta che qualcuno vi propone l'acquisto di software copiato, meditate un attimo al danno provocato da tale gesto e forse, in futuro, allo stesso prezzo potrete comprare una copia originale del programma.

Pietro Dell'Orco

Per collaborare a SUPERVIC

La maggior parte dei seguenti suggerimenti ha lo scopo di migliorare l'accuratezza e la velocità di pubblicazione di un articolo; rispettando questi consigli si accresceranno le probabilità che un vostro lavoro venga pubblicato. La rivista è interessata ad articoli e programmi riguardanti la linea Sinclair. Siamo più interessati al contenuto di un articolo piuttosto che al suo stile, e soprattutto gli articoli devono essere chiari ed esaurienti. La seguente guida permetterà che le vostre buone idee e i vostri programmi vengano più facilmente accettati per la pubblicazione:

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale e data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere il nome del computer al quale il lavoro si riferisce, unitamente alla configurazione richiesta (memoria occorrente, eventuali periferiche e così via).

3 il titolo sottolineato dell'articolo dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina progressivo. Per esempio, Horace Goes.../Brambilla/2.

5 tutte le linee del testo dell'articolo dovranno essere battute con spazio 2 o spazio 3, e un margine di circa un centimetro dovrà trovarsi ad entrambi i lati dello scritto.

6 dovrà essere usata una carta formato A4 e lo scritto dovrà occupare un solo lato del foglio (caratteri maiuscoli e minuscoli).

7 i fogli dovranno essere uniti con una clip.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o la cartuccia per Microdrive dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le

eventuali espansioni richieste. Come suggerimenti di programmazione, si consiglia di usare, per esigenze di stampa listati, le istruzioni INK, PAPER, INVERSE piuttosto che scrivere direttamente in INVERSE VIDEO. Un rapido controllo dei programmi per operare queste sostituzioni sarà da noi estremamente apprezzato.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT etc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza — da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere fotografie, queste dovranno essere in formato 24 x 36, o 6 x 6, in bianco e nero o diapositive.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del programma (da un minimo di L. 50.000 a un massimo di L. 300.000). Il pagamento è effettuato in caso di pubblicazione del lavoro.

15 il materiale ricevuto e non pubblicato non verrà restituito.

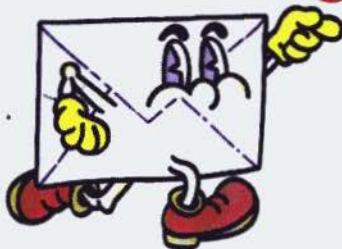
Spedite i vostri lavori a:

SUPERSINC
Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

&WRITE



La posta
dei lettori

Super expander e Ok-list

Usando il programma Ok-LIST apparso sul primo numero della rivista (a proposito complimenti!) sul mio VIC 20 con la cartridge Super Expander inserita, succedono cose strane se uso il comando Key o comunque uno qualsiasi dei tasti di funzione: il computer non risponde ed a volte si blocca completamente. È normale che ciò accada?

Guido Traccanella

R. Non c'è assolutamente nulla che non funzioni nella tua cartridge Super Expander: semplicemente l'OK-LIST non può essere usato se tale cartridge è inserita nel computer. Ci sono dei conflitti nelle zone di memoria impiegate sia dalla Super Expander che dall'OK-LIST che provocano il blocco del VIC. Benché la Super Expander sia incompatibile con OK-LIST, molte altre cartucce di espansione non creano invece alcun problema. Il sistema più sicuro è quindi staccare la Super Expander quando usi OK-LIST.

Seriale o parallela

Che cosa si intende per stampante seriale o parallela?

Roberta De Bernardi

R. Una stampante può ricevere ed inviare dati al computer, e quindi necessita di una INTERFACCIA (connessione che permette a due apparati diversi di comunicare tra loro), ed alcune stampanti posseggono già di per sé sia l'interfaccia seriale che quella parallela. Una stampante seriale ha un solo canale e riceve un bit alla volta — in una serie — dal computer, mentre una parallela ha un connettore multicanale e riceve un byte (otto bit) alla volta. Le stampanti parallele sono quindi più veloci, più facili da usare con una gran varietà di software, ma più costose delle seriali. Queste ultime per contro richiedono

spesso all'utente di dover manipolare certe funzioni (come numero di BAUD, grandezza della parola, etc.) per ottenere la compatibilità con il software usato.

A proposito di compatibilità...

Caro SuperVic, chiedo ospitalità per affrontare un argomento dolente: la compatibilità tra il disk drive 1540 ed il C64, per il quale sembrerebbe indispensabile il drive 1541. In realtà il problema può essere ovviato con la seguente semplice routine:

```
POKE53265,11:LOAD"Nome-  
file":POKE53265,27
```

La prima POKE disabilita il video, la seconda lo ripristina. L'acquisto di un nuovo drive è l'ostacolo che frena molto "vicentisti" dal passare al C64; io ero uno di questi, fino a che non ho scoperto l'arcano. Infatti sto scrivendo con il mio nuovo C64, con il Magic Desk caricato da disco tramite il 1540.

Se poi si vuole rendere il 1540 uguale al 1541, si deve procedere nel solito modo, cioè sostituire la ROM P/N 325303-01 con la ROM P/N 901220-02, ammesso e non concesso che la Commodore voglia fornirli, ed i centri di assistenza sappiano di che cosa si parla e vogliano provvedere. Come ultima spiaggia rimane... la compiacenza di un amico possessore di 1541 che abbia il coraggio di prestarcelo: provveremo a fare una copia della citata ROM, grazie alle diaboliche macchinette oggi esistenti, su una appropriata EPROM... ed il gioco è fatto.

Romano Cappelletti

R. Ringraziamo a nome dei lettori "1540ntisti" il sig. Cappelletti per la preziosa informazione.

MLX e salvataggio su disco

Ho battuto il programma MLX. Tutto funziona perfettamente, solo che quando all'inizio di un riga batto SHIFT + S per sal-

vare quanto ho scritto mi compare regolarmente la scritta "nome file?" e l'altra "N o D?". Batto la D per il disco, ma il floppy alla fine lampeggia senza aver registrato nulla. Come mai?

Luciano Guccini

R. Ciò è del tutto naturale se si cerca di salvare su disco un file usando un nome che già esiste sulla directory del disco. Se si vuole eliminare questa "protezione", basta aggiungere la chiocciolina del "replace" nella linea 750 di MLX, modificandola come segue:

```
750DV = 1.7*(A$ = "D")/FDV =  
8THENF$ = "C0:" + F$
```

Con questa modifica, MLX registrerà su disco un dato file contrassegnato da un certo nome, sostituendo un eventuale file con lo stesso nome presente sul disco.

Cold start e sys

Sono un utente del VIC 20 e, casualmente, ho scoperto che battendo SYS64802 e premendo RETURN si ottiene la cancellazione della memoria senza spegnere l'apparecchio: vorrei sapere se questo procedimento può danneggiare il VIC.

Andrea Giovannetti

R. All'indirizzo decimale 64802 inizia nel VIC (nel C64 la locazione è la 64738) la routine del sistema operativo di cold start. Questa routine viene eseguita automaticamente ogni volta che si accende la macchina, ed assolve il compito di inizializzare molti puntatori di inizio e fine memoria, i vettori di input/output, lo schermo e la tastiera, viene controllata la presenza o meno di una cartidge e, se presente e dotata di autostart, viene passato il controllo a quest'ultima. L'effetto "esterno" di una SYS a questo indirizzo è, dopo un breve intervallo di tempo necessario a compiere tutte le operazioni di cui sopra, l'apparizione del titolo iniziale "Commodore Basic Version V2, etc.", e l'azzeramento della RAM pressoché totale. Questa procedura, ben lontana dal danneggiare in qualunque modo il VIC o il C64, è anzi da preferirsi rispetto allo spegnimento ed alla riaccensione del computer stesso.

La piccola guida del principiante

Che cos'è un programma?

Il solo computer non può compiere alcuna operazione. Un computer possiede potenzialità ma, come una macchina senza benzina, senza un programma, non può funzionare.

La maggior parte dei programmi pubblicati su SUPERVIC per i computer Commodore sono scritti in un linguaggio per calcolatori chiamato BASIC. Il BASIC è facile da imparare ed è disponibile, di serie, nel VIC 20 e nel Commodore 64.

Programmi BASIC

Ogni mese, SUPERVIC pubblica programmi sia per il VIC che per il 64. Tanto per cominciare, se possiedi un VIC copia solo i programmi scritti per la tua macchina indicati con "versione per VIC 20". Più tardi, quando avrai acquisito esperienza con il BASIC del tuo computer, potrai cercare di copiare e convertire certi programmi per altri calcolatori. Diversamente dal linguaggio corrente, che può essere variamente interpretato, il BASIC di solito ha un solo modo corretto per indicare qualcosa.

Ogni lettera, carattere, o numero ha il suo significato. Un errore banale è costituito dalla sostituzione nel numero "0" con la lettera "O" oppure il carattere minuscolo "l" invece del numero "1" o ancora il carattere minuscolo "B" con il numero "8".

Devi anche inserire tutta la punteggiatura, i due punti (:) e le virgole copiando esattamente ciò che appare sulla rivista. Gli spazi possono essere importanti. Per essere sicuro, copia il listato esattamente come si presenta.

Le parentesi e i caratteri speciali

L'eccezione per questa regola di copiatura si presenta quando incontrerete indicazioni tra parentesi quali: "(GID)".

Ogni cosa compresa tra parentesi è un carattere speciale oppure un carattere che non può essere facilmente prodotto con la stampante. Incontrando un carattere di questo tipo fate riferimento alla "Piccola guida per l'input" dei programmi.

Le istruzioni DATA

Alcuni programmi contengono una sezione, o delle sezioni, di istruzioni DATA. Queste linee di istruzione forniscono le necessarie informazioni di cui il programma ha bisogno.

In alcuni casi le istruzioni DATA costituiscono il programma vero e proprio, altri contengono codici grafici. Queste

linee sono particolarmente soggette agli errori. Se un solo numero in una linea di istruzione DATA è sbagliato il calcolatore potrebbe "plantarsi" o distruggere il programma. La tastiera e il tasto STOP appaiono inattivi e lo schermo completamente vuoto. Non lasciarti prendere dal panico. Non si è verificato alcun danno.

Per riprendere il controllo devi spegnere il computer e successivamente riaccenderlo. Ciò cancellerà qualsiasi programma presente in memoria per cui è necessario sempre fare il SAVE del vostro programma prima di comandare il RUN.

Se il computer si ferma puoi caricare (LOAD) il programma e cercare l'errore.

A volte, quando il programma viene "lanciato", un'istruzione DATA errata può causare un messaggio di errore.

Il messaggio di errore potrebbe riferirsi alla linea di programma che legge (READ) il contenuto delle istruzioni DATA.

Come conoscere il computer

Dovresti prendere confidenza con il computer prima di procedere alla copiatura del programma.

Impara le istruzioni che si usano per memorizzare e richiamare i programmi da nastro o da disco. Dovrai conservare una copia del tuo programma se non vorrai copiarlo ogni volta che lo devi usare. Impara ad usare le funzioni di "editing" della tua macchina. Come puoi correggere un errore? Puoi sempre ricopiare la linea e in questo caso devi sapere come procedere. Sapresti come inserire i caratteri in "inverse", i caratteri minuscoli e quelli di controllo? Tutto ciò è spiegato nel manuale del calcolatore.

Un veloce ripasso

- 1) Copiare il programma una linea alla volta, con ordine. Premere RETURN alla fine di ogni linea. Usare il tasto "DEL" per correggere gli errori.
- 2) Confrontare la linea copiata con quella presente nella rivista. Puoi controllare l'intero programma nel caso in cui si presenti un errore quando esegui il RUN del programma.
- 3) Accertati di avere inserito le istruzioni tra parentesi graffe con gli appropriati caratteri di controllo (fai riferimento alla "Piccola guida per l'input" dei programmi) che trovi in questa stessa rivista).

Siamo spiacenti di non poter rispondere alle singole richieste di informazioni circa i programmi, prodotti, o i servizi apparsi su SUPERVIC.



OPEN



a cura della **Redazione**

Commodore 16

Il nuovo Commodore 16 è stato progettato appositamente per chi si avvicina per la prima volta al mondo dell'informatica, ed è perciò semplice da usare e da programmare.

La tastiera di dimensioni standard è la stessa del VIC 20 e del Commodore 64. Oltre ai normali tasti di funzione, il Commodore 16 dispone di quattro tasti per il controllo del cursore, e di un tasto di Help.

Il linguaggio Basic versione 3.5, incorporato nella memoria ROM, è stato arricchito e comprende oltre 75 comandi per la gestione completa delle capacità grafiche e sonore, e per la programmazione avanzata nel linguaggio per computer più famoso del mondo. Il tasto Help si dimostra utilissimo in fase di sviluppo dei programmi Basic, perché permette di evidenziare le linee contenenti gli errori di sintassi, individuando le singole istruzioni inesatte. Il Commodore 16 è anche un superbo videogioco. Con i due ingressi per joystick, la grafica in alta risoluzione a 121 colori e i due generatori di suono incorporati, diventa un eccezionale videogioco.

CARATTERISTICHE TECNICHE

Memoria

RAM 16K standard di cui 12K utilizzabili in Basic dall'utente. ROM 32K con sistema operativo e interprete Basic incorporati.

Microprocessore

7501 con clock da 0.89 a 1.76 MHz.

Video

40 colonne per 25 righe di testo.



Colori

121 colori (15 colori di base con 8 gradazioni più il nero).

Set di caratteri

Lettere maiuscole e minuscole, numeri e simboli.
Caratteri in negativo e lampeggianti.
Set di caratteri grafici Pet Commodore.

Display mode

Caratteri di testo.
Grafici in alta risoluzione e in multicolor.
Testo in schermo diviso/grafici in alta risoluzione o in multicolor.

Risoluzione grafica

320x200 pixel

Suono

2 generatori di suono oppure 1 generatore di suono e un generatore di rumore bianco. 9 livelli di volume per le due voci.

Tastiera

Standard qwerty con 66 tasti.
4 tasti di controllo del cursore.
4 tasti di funzione programmati, che possono essere ridefiniti dall'utente per supportare fino a 8 funzioni predefinite. Tasti di controllo del colore.
Tasto di Help.
Set di caratteri maiuscolo e minuscolo.
Set di caratteri grafici.

Input/Output

User port C16
Porta seriale Commodore.
Porta di collegamento per unità registratore C1531
Uscita monitor con segnale composito/croma/luma.
Input/Output audio.
Ingresso alimentazione di rete.

Caratteristiche principali

Basic esteso versione 3.5 incorporato, con oltre 75 comandi in grado di pilotare grafica e suono.
Monitor in linguaggio macchina incorporato, con 12 comandi disponibili.
Possibilità di creazione finestre su video.

Periferiche opzionali

SFS 481 unità a disco veloce
C 1542 unità a disco
MCS 801 stampante a matrice a 7 colori
MPS 802 stampante a matrice
DPS 1101 stampante a margherita 'letter quality'
C 1520 stampante/plotter a 4 colori
C 1703 monitor a colori

Altre periferiche

Commodore 16 funziona anche in collegamento con l'unità a disco C 1541, con la stampante a matrice C 1526 e con il monitor a colori C 1701.

Dimensioni

Altezza: 76.2 mm
Larghezza: 406.4 mm
Profondità: 203.2 mm

Alimentazione di rete

220 volts AC, 50 Hz con alimentatore esterno.

Consumo

8.5 watt massimo.

Colore

Grigio antracite.

Commodore PLUS/4



Il suo nome deriva dai quattro programmi applicativi incorporati nel computer ed immediatamente disponibili al momento dell'accensione. Il software è contenuto nella memoria ROM del Commodore Plus/4, e si richiama con la semplice pressione di un tasto.

Il software integrato del Commodore Plus/4 comprende le applicazioni più diffuse e più utilizzate nell'attività quotidiana: un programma di word processing per scrivere lettere e relazioni, un foglio elettronico per la pianificazione finanziaria, un database per la creazione e la gestione di un archivio, e un pacchetto di business graphics per visualizzare sotto forma di diagrammi e istogrammi i dati memorizzati.

File manager permette di raccogliere, memorizzare, riordinare e richiamare informazioni di qualsiasi genere: dagli indirizzi per una mailing list agli articoli di un inventario, agli elementi di un elenco personalizzato, alle ricette di cucina. Il programma offre la possibilità di riordinare e riorganizzare le informazioni raccolte per creare lettere personalizzate, etichette per spedizioni, rapporti e tabelle. Inoltre il File Manager è uno strumento flessibile adatto per qualsiasi applicazione personale, e non costringe chi lo usa a imparare complicate tecniche di programmazione per ottenere operazioni complesse. Spreadsheet è un foglio elettronico incorporato che esegue automaticamente calcoli su dati numerici soggetti a continue variazioni. È l'ideale per preparare proiezioni finanziarie, bilanci familiari o aziendali, analisi di budget e per la pianificazione in generale.

Ogni volta che un dato varia, il computer ricalcola istantaneamente tutti i dati correlati, fornendo i nuovi valori per l'intero foglio di lavoro.

Wordprocessor permette di creare lettere, relazioni, rapporti e documenti scritti di qualsiasi tipo con grande facilità. I testi possono essere memorizzati su disco e corretti più volte prima di procedere alla stampa su carta.

Il programma è collegato con lo Spreadsheet e può richiamare tabelle di dati da includere in un documento scritto. Questa perfetta integrazione fra programmi incorporati è un concetto rivoluzionario nel settore degli home computer.

Graphics è anch'esso incorporato nel Commodore Plus/4 ed è integrato con il foglio elettronico. Permette di visualizzare sotto forma di grafico i dati inseriti nello spreadsheet, per realizzare un sistema completo di business graphics. 12 comandi Basic aggiuntivi rendono semplicissima la creazione di programmi grafici personalizzati, che possono essere memorizzati su nastro o su disco. Con Graphics il Plus/4 è in grado di disegnare cerchi, rettangoli e complesse figure geometriche, colorarle a piacere e includerle nei testi scritti con il word processor.

I programmi incorporati possono dividere e scambiare le informazioni memorizzate dal computer. Inoltre, l'opzione di screen window permette di utilizzare due programmi contemporaneamente, visualizzandoli entrambi sullo schermo.

Queste caratteristiche uniche fanno del Commodore Plus/4 la macchina ideale per l'attività professionale. Ma non solo. I computer si stanno diffondendo nel mondo della scuola, e possono essere utilizzati con profitto anche a casa. Il Plus/4 ha tutte le caratteristiche per costituire lo strumento ideale per lo studente. Il Basic esteso incorporato nella ROM offre oltre 75 comandi di programmazione, compresi quelli per gestire grafica e suono. I programmi incorporati possono dividere e scambiare le informazioni memorizzate dal computer.

In ambiente domestico, le possibilità di utilizzo del Plus/4 sono praticamente infinite. Grazie al software integrato, computerizzare il bilancio familiare la corrispondenza personale e l'archiviazione di dati diventa semplicissimo. In aggiunta al software sviluppato espressamente per questo computer, Commodore Plus/4 può utilizzare anche i programmi scritti per il Commodore 16.

CARATTERISTICHE TECNICHE

Memoria

64K Ram standard di cui 60K utilizzabili in Basic dall'utente.

ROM

32K ROM standard (compresi sistema operativo e interprete Basic)

Video

40 colonne per 25 righe di testo

Colori

121 colori (15 colori di base con 8 gradazioni più il nero)

Set di caratteri

Lettere maiuscole e minuscole, numeri e simboli.
Caratteri in negativo e lampeggianti.
Set di caratteri grafici Pet Commodore.

Display mode

Caratteri di testo.
Grafici in alta risoluzione e in multicolor.
Testo in schermo diviso/grafici in alta risoluzione o in multicolor.

Risoluzione grafica

320x200 pixel

Suono

2 generatori di suono oppure 1 generatore di suono e un generatore di rumore bianco. 9 livelli di volume per le due voci.

Tastiera

Standard qwerty con 67 tasti.
4 tasti di controllo del cursore.
4 tasti di funzione programmati, che possono essere ridefiniti dall'utente per supportare fino a 8 funzioni predefinite. Tasti di controllo del colore.
Tasto di HELP.
Set di caratteri maiuscolo e minuscolo.
Set di caratteri grafici.
Pulsante di Reset del sistema.
Tasto di Escape.

Input/Output

User port.
Porta seriale Commodore.
Porta per cartuccia ROM e per unità di disco parallela.

2 porte per Joystick.
Porta di collegamento per unità registratore C1531.
Uscita monitor con segnale composito/croma/luma.
Input/output audio.
Ingresso alimentazione di rete.

Caratteristiche principali

Basic esteso versione 3.5 incorporato, con oltre 75 comandi in grado di pilotare grafica e suono.
Monitor in linguaggio macchina incorporato, con 12 comandi disponibili.
Possibilità di creazione finestre su video.
4 programmi applicativi incorporati: File Manager, Spreadsheet, Word Processor, Graphics.

Periferiche opzionali

SFS 481 unità a disco veloce
C 1542 unità a disco
C 1531 registratore a cassette
MCS 801 stampante a matrice a 7 colori
MPS 802 stampante a matrice

C 1520 stampante/plotter a 4 colori
C 1703 monitor a colori

Altre periferiche

Commodore Plus/4 funziona anche in collegamento con l'unità a disco C 1541, con la stampante a matrice C 1526 e con il monitor a colori C 1701.

Dimensioni

Altezza: 66.7 mm
Larghezza: 422.3 mm
Profondità: 238.1 mm

Alimentazione di rete

220-240 volts AC, 50-60 Hz con alimentatore esterno.

Consumo

8.5 Watt massimo

Colore

Grigio antracite

Home Babysitter

La più moderna e paziente bambinaia

Calcolatori, microprocessori e strumenti programmabili stanno riempiendo ogni aspetto della nostra vita, e tra non molto chi non ha dimestichezza con tastiere, video e affini si troverà in una situazione di analfabetismo culturale.

Per avvicinare il prima possibile i bambini al mondo dei calcolatori (oltre che per dare una mano ai genitori nel loro difficile compito di educatori) la Commodore ha realizzato per il VIC 20, un programma chiamato "Home Babysitter" che, come dice appunto il nome, intrattiene i bambini con passatempi divertenti ed educativi.

Essendo il programma su cartuccia ROM, tutto quello che bisogna fare per cominciare è inserire la cassetta e accendere il VIC 20.

Fatto questo basta assicurarsi che l'alimentatore sia fuori portata per poter lasciare il bambino addirittura solo con il computer; anche se da un punto di vista psicologico il primo approccio dovrebbe essere guidato da un adulto, soprattutto nel caso di bambini molto piccoli.

Essendo rivolto a bambini dai quattro ai sette anni, Home Babysitter fa molto uso delle possibilità grafiche e sonore del VIC 20, allo scopo di catturare l'attenzione dei suoi piccoli "utenti".

Accendendo il calcolatore viene subito presentato, come in tutti i programmi che si rispettano, il "menù", che però ha la caratteristica di essere a colori,

animato e accompagnato da suoni vari.

Dal punto di vista operativo Home Babysitter Commodore offre tre possibilità, selezionabili tramite i tasti funzione, e precisamente: giocare con l'alfabeto, giocare con i numeri e disegnare una faccina.

Se si sceglie l'opzione alfabeto, sullo schermo appaiono, al ritorno di una canzoncina, dei cubetti (versione 2000 dei vecchi cubi di legno) con disegnate le lettere dell'alfabeto.

Dopo la musica il VIC 20 chiede al bambino di ripetere la sequenza.

Ogni tasto premuto esattamente viene premiato con una nota di quella canzoncina ascoltata prima, cosicché il bambino, dopo qualche tentativo e magari con l'aiuto di un adulto, riesce a suonare tutto il motivetto ed a imparare, senza accorgersene, le lettere dell'alfabeto.

Il gioco con i numeri è graduabile in diversi livelli di difficoltà e, a seconda del livello scelto, il computer presenta sullo schermo un certo numero di figurine (ochette, pupazzi, eccetera), chiedendo poi al bambino quale sia il numero complessivo.

In caso di risposta esatta, mentre viene segnato un punto a favore, appare una faccina sorridente accompagnata da una musicchetta allegra. In caso di errore, invece, appare una faccina arrabbiata e il VIC chiede al bambino di riprovare.

In caso di nuovo errore il programma "riconta" le figure una per una davanti al bambino e segna un punto di demerito.

Alla fine della serie, se il risultato è stato soddisfacente, compare di nuovo la faccina contenta.

Nell'ultima parte del programma, il VIC 20 disegna una faccina buffa, e tramite i tasti di funzione è possibile cambiare a questa figura i capelli, gli occhi, il naso, la bocca e le orecchie. Il tutto scegliendo anche uno qualsiasi degli otto colori che il VIC stesso può rappresentare sullo schermo.

Ovviamente le combinazioni possibili sono moltissime e il bambino, una volta imparato come fare, rimarrà per lungo tempo davanti al calcolatore, interrompendosi solo per mostrare alla prima persona disponibile un disegno particolarmente buffo o riuscito.

Home Babysitter è in vendita a L. 41.000 + IVA



Piccola guida per l'input dei programmi

Molti dei programmi listati da **SUPERVIC** contengono particolari caratteri di controllo (controllo cursore, tasti colore, video-inverse, etc.).

Per rendere chiaro ciò che dev'essere battuto quando è necessario inserire uno dei suddetti caratteri sono state stabilite le seguenti convenzioni.

Generalmente i listati per VIC 20 e C64 contengono delle «parole» racchiuse tra parentesi graffe (); tali parole rappresentano particolari caratteri di controllo: {GIÙ} significa premere il simbolo del cursore verso il basso. {5 SPAZI} vuol dire battere 5 volte la barra-spazio.

Per indicare che un tasto dev'essere «shiftato», cioè premuto insieme al tasto SHIFT, nel listato il simbolo di quel tasto sarà sottolineato.

Per esempio S significa che dev'essere premuto il tasto S mentre è abbassato il tasto SHIFT.

La manovra farà apparire sullo schermo un piccolo «cuore».

Trovando un simbolo sottolineato chiuso tra parentesi graffe (esempio {10 N}) dovrà essere interpretato come «premere il carattere indicato per il numero di volte che lo precede nelle parentesi» nel nostro esempio premere 10 volte il tasto N «shiftato».

Se il tasto da premere è indicato tra le parentesi () vuol dire che lo stesso dovrà essere premuto mentre è premuto il tasto «Commodore» (il tasto «Commodore» è quello posto nell'angolo in basso a sinistra). Se il tasto scritto tra () è preceduto da un numero, ciò significa che il tasto dev'essere premuto per il numero di volte indicato.

Raramente si potrà trovare un carattere alfabetico solitario racchiuso tra parentesi graffe. Tale carattere sul C64 può essere battuto mentre è premuto il tasto CTRL.

Ad esempio {A} sta ad indicare la sequenza CTRL-A. A proposito del «modo virgolette» è nota la possibilità di muovere il cursore sullo schermo con i tasti CRSR. Ogni tanto i programmatori desiderano muovere il cursore durante l'esecuzione del programma. È per questo motivo che nei programmi si troveranno dei {SIN}, {HOME} e {BLU}.

L'unico modo perché il calcolatore distingua il comando diretto da quello inserito nel programma è il «modo virgolette». Premendo il tasto «virgolette» (tasti SHIFT 2) il calcolatore si predispose appunto in «modo virgolette».

Battendo un carattere qualsiasi cercando poi di correggerlo muovendo il cursore a sinistra, si otterrà solo un tratto verticale in un quadratino inverso. Questo è infatti il simbolo del cursore a sinistra nel «modo virgolette».

L'unico comando di editing non utilizzabile all'interno di un programma è il DEL. Battendo nuovamente le «virgolette» il calcolatore lascia il «modo virgolette». Si può accedere al «modo virgolette» quando inserite spazi nella linea.

In ogni caso, il sistema più semplice per uscire dal «modo virgolette» è quello di premere il tasto RETURN.

Utilizzare la tabella che segue quando è necessario inserire comandi relativi al cursore e ai tasti colore.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR HOME		{CYN}	CTRL A		{<7>}	CTRL 7	
{HOME}	CLR HOME		{PUR}	CTRL B		{<8>}	CTRL 8	
{SU}	SHIFT [] CRSR []		{GRN}	CTRL C		{F1}	CTRL 1	
{GIU'}	[] CRSR []		{BLU}	CTRL D		{F2}	CTRL 2	
{SIN}	SHIFT [] CRSR []		{YEL}	CTRL E		{F3}	CTRL 3	
{DES}	[] CRSR []		{<1>}	CTRL 1		{F4}	CTRL 4	
{RVS}	CTRL N		{<2>}	CTRL 2		{F5}	CTRL 5	
{OFF}	CTRL O		{<3>}	CTRL 3		{F6}	CTRL 6	
{BLK}	CTRL 1		{<4>}	CTRL 4		{F7}	CTRL 7	
{WHT}	CTRL 2		{<5>}	CTRL 5		{F8}	CTRL 8	
{RED}	CTRL 3		{<6>}	CTRL 6				

Iniziare con un sistema a dischi Parte seconda: primi passi

di C. Brannon
trad. e adatt. di M. Cristuili Grizzi

Dopo una breve analisi del motivo per cui è necessario fare delle copie di back-up dei dischi più importanti - e del perché alcuni dischi non possono essere copiati - vedremo come iniziare ad usare appropriatamente il vostro drive 1541.

Il mese scorso abbiamo visto perché è così importante produrre delle copie di back up dei propri dischi: poiché un disco può contenere una così elevata quantità di dati (più di 170000 caratteri), c'è molto da perdere se gli accade qualcosa di spiacevole. Supponiamo abbiate fatto una copia di un programma applicativo, come un buon word processor; potete usare correntemente il programma sul disco di back up, mentre il disco originale è stato riposto in un luogo sicuro da tutti quei fattori che, come abbiamo visto nella scorsa puntata, possono danneggiarlo. Questa sembra una procedura talmente ovvia che molti stupiscono quando scoprono invece che molte aziende produttrici di software proteggono i propri dischi contro la duplicazione. Il fatto è che i produttori di software proteggono i propri dischi dalle copie illegali e per ottenere questo scopo, purtroppo, inibiscono anche la onesta duplicazione di un programma a fini cautelativi. Del resto si pensi al mancato introito che deriva a queste aziende dal diffondersi di copie abusive di programmi originali, vendute ad un prezzo molto minore di quello originario. Tempo fa, il lavoro di un programmatore era tangibile ed unico, mentre ora che i computer sono diventati quasi l'equivalente informatico di una fotocopiatrice, le cose si sono alquanto modificate. Come si può stabilire un prezzo adeguato per un'automobile, quando se ne può produrre una copia atomo per atomo, con l'energia elettrica come unico ingrediente? Questa af-



fermazione suona del tutto fantastica, ma nel campo dell'informazione siamo giunti a questo livello. I cosiddetti programmi originali non hanno più valore delle copie da essi prodotte: la sola differenza tra un disco vergine ed un word processor da duecentomila lire sta in una organizzazione di campi magnetici fantasma su un disco magnetico dal costo di cinquemila lire... Con il software così facile da copiare, non è facile combattere la pirateria. Il disk drive è stato costruito per trasformare i campi magnetici letti sul dischetto in numeri che possono essere usati dal computer. La protezione dalla

copia permette questo trasferimento, ma cerca anche di impedire che il disco venga letto al di fuori dalla specifica applicazione per la quale è stato prodotto. I metodi usati per le protezioni sono complessi quanto la meccanica del drive possa permettere, ma sono in realtà efficienti solo per prevenire casuali istruzioni di LOAD o SAVE o copie di files dati. Inoltre a volte il sistema di protezione del disco è così sensibile che anche la copia originale non "gira" se il vostro disk drive è leggermente fuori allineamento. Le aziende devono proteggere il proprio software, ma che dire allora dell'utente che desi-

Iniziare con un sistema a dischi parte seconda: primi passi

derà solamente prodursi una back up copy? Molte aziende offrono in garanzia un disco nuovo se quello acquistato presenta dei difetti, ma se quel software vi è indispensabile per il lavoro o per qualsiasi altro motivo, chi vi ripaga delle settimane passate ad aspettare di ricevere la copia sostituita in garanzia?

In teoria, ogni computer dovrebbe avere un particolare numero di serie leggibile via software, in modo che quando si usi per la prima volta un certo programma commerciale che si è acquistato, questo possa leggere il numero di serie del vostro computer, confrontarlo con quello contenuto nel programma e precedentemente inserito dal rivenditore, e, se questi numeri coincidono, produrre automaticamente una copia di se stesso che potrete tenere come copia di back up. Chiunque altro usasse il vostro programma con un diverso computer, causerebbe una discrepanza nei numeri di controllo, provocando il blocco del programma stesso. Purtroppo, la produzione a larga scala di computer con un siffatto numero di serie distintivo è antieconomica per le aziende produttrici di hardware.

Forse la soluzione migliore è quella già in uso in alcuni casi: il software viene fornito insieme ad una "chiave" elettronica che va collegata al computer in un particolare modo: sul VIC e sul C64 generalmente questa chiave va inserita nella porta del joystick (se non già utilizzata) oppure nella porta di interfaccia con il registratore a cassette. Altri tipi di chiave possono essere dei chip ROM che devono essere installati sugli slot di espansione; il programma così protetto non potrà essere eseguito se la chiave non è stata installata nel posto giusto, ma si potranno produrre quante copie si vogliono del programma stesso. Questo copie non saranno però sfruttabili commercialmente in quanto anch'esse necessiteranno della presenza della chiave per poter essere eseguite, e di chiave se ne possiede una soltanto.

La vendita di software su cartridge è la meno esposta a rischi e di copie pirata, in quanto ancora pochi possiedono la tecnica di duplicazione di una cartridge. Fino a che non verranno emanate precise leggi che proteggano i produttori dalla pirateria del software, si assisterà alla continua gara tra chi realizza protezioni sempre più sofisticate e chi per contro elabora metodi di duplicazione sempre più articolati.

Se siete alla prima esperienza con un

disk drive, dovrete per prima cosa imparare a trattarlo con molti più riguardi del vostro vecchio e robusto registratore a cassette: i drive sono delicati strumenti di precisione. Trattatelo con cura quando lo portate a casa, ed ogni qual volta dobbiate spostarlo dal luogo dove abitualmente si trova: non sottoponetelo a scuotimenti, colpi o vibrazioni, in quanto ogni shock fisico può portare le testine del drive fuori allineamento, causando problemi nel salvataggio o nel caricamento dei programmi, con l'inevitabile finale ricorso ad un tecnico esperto che sia in grado di riallineare.

Compresi nella confezione del drive troverete il cavetto DIN di connessione con il computer, il cavo di alimentazione, il manuale d'uso ed un disco dimostrativo. Non dimenticate di acquistare una confezione di dischetti vergini per i vostri primi esperimenti. Probabilmente vorrete dare un'occhiata al disco dimostrativo ed eseguire il test delle prestazioni del drive, ma il manuale non vi aiuterà molto se siete novizi, in quanto è piuttosto oscuro ed incompleto. Così, prima di fare qualunque cosa con il nuovo drive, leggete con un po' di pazienza il resto di questo articolo. Se non avete la minima esperienza con i drive, la prima cosa che vorrete imparare è probabilmente come caricare un programma. In questo caso dovrete già disporre di un disco contenente dei programmi, siano essi giochi piuttosto che un word processor; il software commerciale ben documentato conterrà nel proprio manuale tutte le procedure per il caricamento descritte passo-passo. Usualmente, dovrete digitare:

```
LOAD "*"*,8
```

si accenderà la spia luminosa di colore rosso del drive, il motore girerà e, se ogni cosa funziona correttamente, comparirà dopo poco il messaggio READY. Ora potete dare il comando RUN.

Il tipo di comando LOAD che avete dato ha detto al drive (che è un apparato intelligente) di cercare e caricare in memoria il primo programma presente sul disco. L'uso dell'asterisco sarà spiegato in seguito. Altre volte invece vorrete caricare uno specifico programma dal vostro disco, ed allora userete comandi come:

```
LOAD "INVADERS",8
```

oppure:

```
LOAD "EASY SCRIPT",8
```

Se avete finora seguito esattamente questa procedura ed il programma non vuole saperne di essere caricato in memoria, sarà opportuno verificare eventuali errori. Per sapere se si sono verificati errori, battere questa linea di programma e date il RUN:

```
10 OPEN 15,8,15: INPUT #15,
EN,EM$: PRINTEN: EM$: CLOSE 15:
END
```

Il messaggio di errore ricevuto potrà sembrare troppo codificato, ma può aiutare; in ogni caso se ne parlerà più avanti.

Se ottenete un messaggio di errore, cercate di rimediarvi: controllate che i cavi di interfaccia ed alimentazione siano stati collegati nel modo corretto, che il disco sia stato inserito correttamente e dalla parte giusta, e la porticina chiusa dolcemente, etc. Se tutto è a posto ed ancora si hanno problemi nel caricamento, può essere che il disco sia rovinato, od addirittura sia incompatibile con il vostro drive (ogni drive è leggermente diverso dagli altri in fatto di allineamento e velocità, anche se tutti dovrebbero stare nelle tolleranze definite dal costruttore).

Caricare un programma è una strada a senso unico: il vero punto forte del disk drive è che può contenere una grande quantità di VOSTRE informazioni, non solo materiale "già confezionato". Infatti, il drive è un apparato di memoria di massa: come con la memoria RAM, potete leggere e scrivere sul disco. Quest'ultimo è più lento della RAM, ma comunemente molto più capiente (170K contro 3.5K su un VIC 20!).

Se avete letto la puntata del mese scorso, vi ricorderete che l'accesso più semplice è quello a livello di settore, dove si possono leggere e scrivere blocchi di 254 caratteri. È proprio così complicato come suona, ma fortunatamente non dovrete mai usare il drive ad un livello così primitivo. Al contrario, il drive ed il vostro computer lavorano insieme e di comune accordo per permettervi di creare dei FILES.

Un file è una sequenza di informazioni, non limitata a 254 caratteri, ed è come una lunga sequenza di numeri. I files possono contenere anche caratteri, ovviamente, dal momento che questi sono rappresentati da numeri. Un file può essere un programma, un listato, questo articolo scritto sul word proces-

sor, o semplicemente un insieme di dati. Ogni file è caratterizzato da un nome, che permette un rapido accesso al file stesso da parte dell'utente, semplicemente comunicandolo al computer. Un nome di file può essere generalmente una qualunque sequenza di caratteri, compresi quelli alfabetici, quelli grafici, i segni di interpunzione, etc. Il nome del file può contenere inoltre un massimo di sedici caratteri. Esempi di nomi validi sono: "PROGRAMMA N.1", "BBASICTOOL", "OK-LIST". Alcuni caratteri sono riservati, come l'asterisco ed il punto interrogativo. Quest'ultimo viene usato come il jolly in certi giochi di carte: quando cerca il nome di un file su un disco, il drive confronta il nome che gli avete richiesto, carattere per carattere, con tutti i nomi di file presenti sul disco: il punto di domanda vi permette qualche ambiguità. Se non siete sicuri che il nome del file che vi ricordate sia proprio quello giusto, potete sostituire dei punti interrogativi ai caratteri dei quali dubitate la certezza. Se siete in dubbio tra "TRIX" e "TRIS", potete battere "TRI?", oppure se non vi ricordate se il nome corretto sia "HAPPY FACE" piuttosto che "NAPPYFACE", potete battere "?APPY?FACE".

Molto più utile è l'asterisco: vi permetterà di "tagliare" dei caratteri. Per esempio, "BAS*" corrisponderà a "BASIC AID", a "BASEBALL", a "BASH", etc. L'asterisco da solo corrisponde a qualunque nome di file, ed è per questo che si usa l'espressione LOAD***.8 per caricare il primo file sul disco; infatti il comando LOAD"BAS***.8" caricherà il primo file incontrato sul disco il cui nome inizia con i caratteri "BAS".

Prima che possiate SCRIVERE su un disco per la prima volta, dovete formattarlo. Alcuni programmi applicativi, come i word processor o certi database, vi permettono di compiere direttamente da programma questa operazione, ma generalmente dovete eseguirlo da BASIC. Un disco vergine appena estratto dalla confezione non è ancora pronto per il vostro disk drive; quest'ultimo non sa dove trovare tracce e settori, dal momento che il disco è solo un pezzo circolare di supporto magnetico. È il drive che deve organizzarlo in tracce e settori, scrivendo informazioni temporizzate su tutta la superficie del disco. E ciò in aggiunta a qualsiasi tipo di dati voi poi scriverete sul disco stesso.

Per inviare un comando al disco, dovete aprire un CANALE DI COMANDO. Dovrete quindi digitare:

OPEN 15,8,15

Questa istruzione dice al computer che userete il canale numero 15 per parlare al drive, apparato numero 8. L'ulti-

mo numero, un altro 15, è per uso del drive: gli dice che le cose che gli inviate sono comandi, non dati. Tutti i comandi vengono inviati con l'istruzione PRINT# (che si pronuncia "print file"). A differenza della solita PRINT usata per scrivere sullo schermo, non si può usare come abbreviazione il punto interrogativo, bensì l'abbreviazione ammessa per PRINT# è P + SHIFT (R). L'istruzione per formattare un disco ha questa struttura:

PRINT# 15, "N:NOME DEL DISCO.ID"

La N sta per NEW, parola chiave che la Commodore usa per la formattazione. Si può anche scrivere così:

PRINT# 15, "N:NOME DEL DISCO.ID"

Questo comando cancella completamente il disco nel caso non fosse nuovo, quindi attenzione! I due punti separano il comando dai parametri che il comando abbisogna per essere eseguito. Il nome del disco usa lo stesso formato del nome del file, e può consistere in qualsiasi cosa di vostro gradimento.

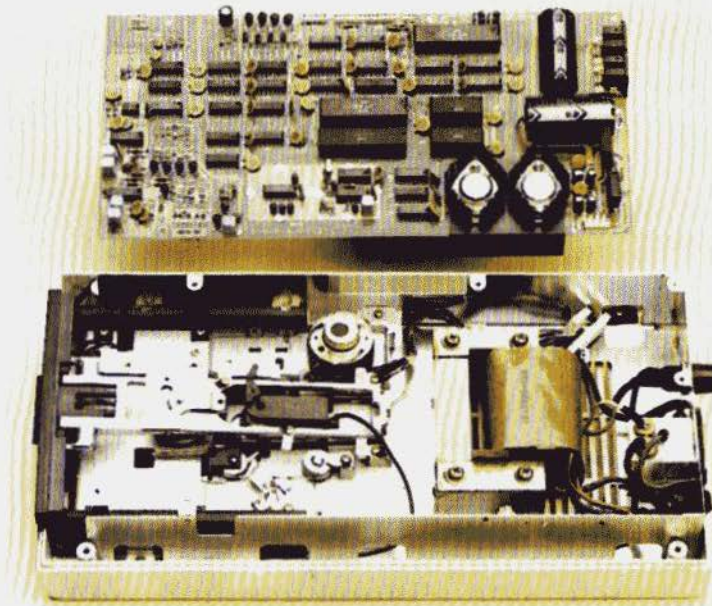
È opportuno avere un minimo di organizzazione: non registrate a caso i file sui vostri dischi, ma preparatene uno per i giochi, uno per le utilities, uno per i vostri programmi BASIC, e così via... Ciò rende molto più semplice trovare il disco contenente il programma che state cercando in un dato momento; il nome del disco dovrebbe appunto descrivere a questo proposito il contenuto del disco stesso.

Il parametro ID (identificazione) è un codice di due caratteri, e non è usato come il nome del disco per i vostri fini di organizzazione, bensì per uso interno del drive. Se ogni disco possiede un diverso codice ID, il drive si accorge quando cambiate il disco inserendone un altro. Potete usare codici ID da 0 a 99, ed è fondamentale che ogni disco abbia un diverso numero di ID. Idealmente, anche i dischi dei vostri amici che usate sul vostro drive dovrebbero avere degli ID sempre diversi: in pratica, siate prudenti e non assegnate all'ID di tutti i vostri dischi il codice 01. Parleremo più diffusamente del codice ID e del perché deve essere unico più avanti in questa trattazione, unitamente a come si possa leggere questo codice direttamente nei vostri programmi. Esiste inoltre una forma del comando NEW che permette di cancellare semplicemente un disco senza riformattarlo: per ottenere ciò usate l'espressione di cui sopra, cambiando (se lo desiderate) il nome del disco, ma tralasciando il codice ID. Cambiando anche quest'ultimo, il disco sarà anche riformattato.

Ora che il disco è pronto per l'uso, vorrete vedere che cosa contiene. Digitate:

LOAD\$*.8

Quando riappare il cursore con la scritta READY, battete LIST. La DIRECTORY (detta anche CATALOG in altri sistemi) è un listato di nomi di file. All'inizio della lista compare il nome del disco ed il codice di identificazione ID. A sinistra di ogni nome di file com-



Ecco come si presenta l'interno del drive 1541

Iniziare con un sistema a dischi parte seconda: primi passi

pare un numero, che rappresenta l'ammontare dei blocchi da 254 caratteri che il file occupa nel disco. Per ottenere una stima di massima, dividete per quattro questo numero, ed otterrete all'incirca il numero di Kbyte occupati da quel dato programma. Un programma di 25 blocchi usa circa 6 Kbyte dei 170 K disponibili nel disco.

A destra di ogni nome di file compare un'etichetta di tre caratteri PRG, SEQ, REL, oppureUSR. Queste etichette vi informano sul tipo del file: le più comuni sono PRG (PRoGramma) e SEQ (dati SEQuenziali). Entreremo in modo approfondito in queste distinzioni quando parleremo di programmazione. L'ultima riga della directory vi informa sul numero di blocchi da 254 caratteri ancora disponibili sul disco: dividete per quattro per ottenere il numero di Kbyte.

Quando listate la directory di un disco appena formattato, vedrete solo il nome del disco, il suo codice di identificazione, e la scritta "664 BLOCKS FREE". Dividendo per quattro, otterrete solo 166 Kbyte: questo perché un certo spazio sul disco viene sfruttato dal drive per immagazzinare appunto la directory ed altre informazioni utili al suo sistema operativo.

Dopo essere stato formattato, il disco è pronto per ricevere programmi e dati. Se siete pronti anche voi, digitate un microprogramma come questo:

```
10 PRINT "QUESTA È UNA PROVA":
GOTO 10
```

Per copiare il programma dalla memo-

ria RAM del computer al disco, usate il comando SAVE. Sarete già familiari con questo comando dopo l'uso del registratore a cassette: l'unica differenza sarà l'aggiunta di una virgola e del numero otto per comunicare al computer che volete usare il disk drive e non l'unità a cassette.

Pensate ad un nome di file, ricordando che non deve superare i 16 caratteri, e battete:

```
SAVE "0:NOME DEL FILE",8
```

Lo "0" è un nuovo comando: serve per il drive costituiti da due unità a disco, delle quali una è contrassegnata da numero 0 e l'altra dal numero 1. Dal momento che probabilmente possedete una singola unità 1540 o 1541, potete tralasciare questo "0:". In ogni caso, dopo avere premuto RETURN, il motore del drive inizierà a girare e contemporaneamente si accenderà la spia rossa. Non estraete mai il disco quando la spia rossa è accesa e state scrivendo sul disco stesso, altrimenti il computer non riuscirà a scrivere per intero il file in questione: ciò non vale invece se state leggendo dal disco. In quanto non si rischia così di perdere irrimediabilmente dei dati, ma solo di perderli temporaneamente.

Quando la luce rossa si spegne, il programma è stato salvato: se invece lampeggia, significa che qualcosa è andato storto. Potete usare il programmino di una linea che abbiamo visto verso l'inizio dell'articolo per leggere il messaggio d'errore, oppure controllare ciò che avete digitato e scovare l'errore da

voi. Anche se tutto è andato liscio, può darsi che vogliate controllare se il programma sia stato correttamente salvato sul disco. Il comando VERIFY è però molto più utile sull'unità a cassette, in quanto questa non può accorgersi di un errore verificatosi durante il SAVE. Per usare il comando VERIFY sul disco, aggiungete il ",8":

```
VERIFY "NOME DEL FILE",8
```

VERIFY opera come LOAD, con la differenza che, invece di caricare nella memoria, VERIFY confronta con la memoria. Finita l'operazione, verrà visualizzato "OK" (buone notizie), oppure "VERIFY ERROR" (cattive notizie). Potete usare l'asterisco per abbreviare il nome del file. VERIFY "*",8 verifica il primo programma sul disco. Ora battete NEW e caricate in memoria il programma:

```
LOAD "NOME DEL FILE",8
```

Se il file non è sul disco che avete nel drive, o se avete usato un nome errato, il computer visualizza "?FILE NOT FOUND". Cercate di capire la causa dell'eventuale errore e, se necessario, battete LOAD "*",8 e listate la directory per vedere quale sia il nome del file corretto.

È tutto per questo mese. Nella prossima puntata vedremo come semplificare l'uso del disco con il DOS WEDGE, ed analizzeremo altri comandi come DELETE e RENAME. Fino ad allora, leggete attentamente il manuale e vedete se le cose che contiene stanno acquistando un po' più di significato.

L'unico mass-media
di strumenti musicali
e audio-registrazione.

strumenti
MUSICALI



**GRUPPO
EDITORIALE
JACKSON**

QUANDO JACKSON NON È SOLO ELETTRONICA E INFORMATICA.

REM:HW

Hardware

di M. Cristulib Grizzi

Tutto quello che avreste voluto sapere sul registratore a cassette...

La "1530 Datassette Unit, model C2N", denominazione ufficiale del comunemente detto "Registratore Commodore", è essenzialmente la prima periferica che si acquista insieme al VIC 20 o al C64. Mentre in genere si arriva ad avere un buon grado di conoscenza del funzionamento del computer, ben poco si sa dell'unità a cassette. Con questo articolo spero di chiarire qualche idea in proposito.

Il Datassette si presenta con un cavo di interfaccia abbastanza insolito per chi era abituato ai soliti jack audio di altri sistemi. Ciò deriva da una scelta fatta dalla Commodore a metà tra il tecnico ed il commerciale: vedremo infatti che questa interfaccia permette caratteristiche non ottenibili da altri registratori per computer, ma allo stesso tempo obbliga l'acquirente a sobbarcarsi la spesa (per altro piuttosto elevata) del registratore originale Commodore invece di poter sfruttare il solito registratore che quasi sempre già si possiede. Lo spinotto di interfaccia che si connette sul lato posteriore della console tra la porta seriale ed il connettore per il drive, è composto da sei contatti. Questi costituiscono le linee demandate rispettivamente al passaggio dei dati per la scrittura, la lettura, il controllo del motore, la rilevazione della pressione di un tasto sul registratore, l'alimentazione a 5 Volt e la massa. Esiste inoltre un cavetto interamente metallico che fuoriesce da un lato dello spinotto: in genere è quest'ultimo che crea le maggiori perplessità negli utenti, in quanto il manuale non vi fa cenno e nemmeno esi-



ste sul retro del computer qualche vite dove poterlo connettere. Questo cavetto, con funzioni di massa, serve solamente per l'uso del Datassette con i computer della serie PET, mentre non

ha alcuna funzione ne' col VIC 20 ne' col C64. Se non si prevede di usare il proprio Datassette con un computer PET, è anzi buona norma armarsi di forbici e tagliarlo alla base, in quanto se venis-

se casualmente a contatto con qualche pin della User Port potrebbe provocare danni permanenti al vostro computer. La linea più interessante (e quella che lo differenzia da altri sistemi meno sofisticati), è quella che permette al computer di accorgersi se è stato premuto qualche tasto sul Datassette. Quest'ultimo possiede infatti al suo interno uno switch che controlla la pressione dei tre tasti contrassegnati da PLAY, REWIND e FAST FORWARD, e la comunica al computer tramite l'apposita linea sull'interfaccia. A causa però di quest'unico switch, il computer non è in grado di distinguere quale tasto sia stato premuto, ma solo se ne sia stato premuto uno qualunque. Inoltre, l'eventuale pressione di un tasto viene rilevata solo se è aperto un canale di comunicazione tra computer e Datassette. È questa la ragione per la quale, come vi sarà certamente capitato, se al messaggio "press play on tape" premete per sbaglio REWIND o FAST FORWARD, lo schermo si cancella ed il computer cerca di caricare dei dati pensando che sia stato premuto il tasto PLAY.

La linea di controllo del motore, invece, si fa carico di fermare il motore del registratore dopo che sia stato effettuato un LOAD o un VERIFY, alla fine di un SAVE, o dopo ogni passaggio di un blocco di dati di un file sequenziale. I tipi di registrazione che si possono effettuare con il Datassette sono fondamentalmente due: file programma e file dati. I primi sono in pratica dei dump della memoria che, in forma binaria, vengono inviati al registratore direttamente dal sistema operativo. Ciò avviene con i comandi SAVE e LOAD. I file dati, invece, sono una sequenza di caratteri inviati al registratore nello stesso formato con il quale appaiono sullo schermo: il formato ASCII. In realtà, sia i file programma che i file dati possono essere usati per registrare programmi o dati, e viceversa.

I file dati sono quasi sempre creati in ambiente BASIC tramite le istruzioni OPEN, CLOSE, PRINT #, INPUT # e GET #. Quando si apre un file dati, occorre specificare nella OPEN il numero di canale, il numero di periferica, ed un indirizzo secondario. Tutto ciò è molto semplice vedendo la tabella che segue:

* canale: può essere compreso tra 1 e 255, ma poiché il Datassette può operare con un solo canale aperto alla volta, conviene usare sempre il numero 1.

* periferica: il numero assegnato dal sistema operativo al Datassette è il numero 1.

* secondario: è un numero che indica al computer se deve leggere o scrivere dei dati, con i seguenti valori:

0: legge dei dati.

1: scrive dei dati.

2: scrive dei dati e pone un separatore in coda al file, che indica la fine del nastro.

In pratica, l'indirizzo secondario 2 è molto poco usato, in quanto capita raramente che un programma debba essere costretto a cercare il file dati che gli interessi per tutta una facciata di cassetta... Si farebbe prima ad usare carta e penna invece del computer!

Il file dati viene trattato come una serie di stringhe le quali, man mano che vengono generate o calcolate dal programma BASIC, vengono poste sequenzialmente nel buffer del registratore. Il buffer del registratore è una particolare area di memoria che inizia alla locazione 828 ed è lunga 192 byte. Quando il buffer è pieno, il gruppo di dati (che sarà quindi di 192 byte), viene inviato al Datassette mentre il buffer si svuota per ricevere altri dati dal programma. È per questo motivo che quando usate dei file dati il nastro si muove "a scatti": tra una fermata e l'altra del motore vengono infatti trasferiti blocchi di dati da 192 byte, quindi c'è una sosta del motore di circa 2 secondi per permettere al buffer di riempirsi (o di svuotarsi, se si stanno caricando dati) prima del trasferimento del blocco successivo di 192 byte.

I dati si registrano con l'istruzione PRINT # seguita dal numero di canale usato nella OPEN e da una lista di variabili numeriche o di stringa (a seconda del tipo di dati che si vogliono registrare). Ogni variabile è opportunamente separata dalle altre da un CHR\$(13); ad esempio:
200RS = CHR\$(13): PRINT # 1, A; RS
AS;RS;BS;RS;CS; RS;DS;RS;ES
Il CHR\$(13), che è il carattere di RETURN, serve per separare convenientemente le variabili di stringa, che altrimenti sarebbero interpretate dal computer come una unica, lunga stringa. Quando invece leggete dei dati, la lettura deve avvenire con una INPUT # che sia esattamente speculare alla PRINT # con la quale è stato creato il file, a parte i CHR\$(13) che qui non servono. Questo per evitare che il computer trovi valori numerici dove si aspettano valori di stringa, e viceversa. Una linea di lettura per il file generato con l'esempio precedente potrebbe essere:
500 INPUT # 1,F,WS,KS, XS,Y5,Z5

La lettura dei dati da un file può anche essere effettuata con l'istruzione GET # seguita dal numero di canale usato nella OPEN. Questa istruzione permette di leggere il file carattere per carattere, ed è molto più lenta della INPUT #, tuttavia permette di leggere senza problemi file totalmente numerici o totalmente di stringa dei quali non si conosca il formato con il quale sono stati registrati. Quando non ci sono più dati da passare, il programma BASIC deve prevedere una istruzione CLOSE seguita dal numero di canale adottato nella OPEN. A questo punto, se il canale era di scrittura, viene scritto sul nastro un particolare segnale di EOF (End Of File) che ha lo scopo di segnalare al computer la

fine dei dati una volta che il file venga letto.

Infatti, quando il computer in lettura trova questo segnale, la variabile riservata ST (status) cambia valore, assumendo il valore 64 (o -128 se si è usato l'indirizzo secondario 2). Per testare da programma se sono finiti i dati da leggere, basterà allora controllare il valore di st. Esempio:

```
510IF ST = 64 OR ST = -128
  THENCLOSE1
```

Come avviene la registrazione

Il sistema di registrazione Commodore è uno dei più lenti della categoria, anche se questa lentezza viene ripagata da un completo sistema di controllo degli errori. Anzi, le due cose sono complementari perché il procedimento è lento buona parte a causa del controllo stesso degli errori.

Quando registrate su nastro un programma o un blocco di dati, in realtà il sistema operativo lo registra due volte in sequenza, inoltre per ogni gruppo di otto byte trasmessi ne inserisce un nono con funzioni di checksum. In fase di lettura, il computer legge i vari gruppi di otto byte e li confronta con i rispettivi checksum, poi confronta il primo programma registrato con il secondo: se trova qualche discrepanza, segnala un errore di caricamento. Un altro sistema di controllo degli errori — e causa di aumento del tempo impiegato — è la testata di 10 secondi che viene sempre scritta prima di iniziare la registrazione dei dati vera e propria. Questo "header" serve a fare in modo che il nastro, precedentemente fermo, superi il transitorio di avviamento e raggiunga la corretta velocità di scorrimento davanti alla testina magnetica. La testata inoltre contiene una sequenza di brevissimi impulsi, usati per sincronizzare correttamente le temporizzazioni delle routines di lettura in funzione della velocità del nastro. Quest'ultima infatti può variare con gli sbalzi della tensione di alimentazione, e sta quindi al sistema operativo testarla e trovare il giusto fattore di correzione da apportare alle temporizzazioni delle sue routines. Dopo questa testata di 10 secondi, o primo "header", ne segue un altro di 192 byte contenente il nome del file e la locazione di partenza e quella finale dei dati contenuti nel file stesso. Come si vede, i controlli sono molti, a tutto vantaggio dell'affidabilità ma, purtroppo, a discapito della velocità.

Un po' di manutenzione non fa mai male

Al contrario del computer che, a parte la tastiera, non ha parti in movimento, il vostro Datassette ne ha molte ed è quindi soggetto ad usura meccanica. Richiede perciò una certa manutenzione per ridurre al minimo questa usura. La parte più soggetta ad usura in un qual-

siasi registratore a nastro è la testina magnetica di lettura/scrittura. Se già non avete familiarità con i registratori, togliete l'eventuale cassetta dal Datasette, premete il tasto EJECT e poi il tasto PLAY. La testina di lettura/scrittura è quella che appare al centro, mentre alla sua sinistra si trova la testina di cancellazione, ed alla sua destra il capstan che trascina il nastro. Le due testine, a stretto contatto con il nastro magnetico sia in fase di lettura che di scrittura, diventano alla lunga deposito di ossidi ed impurità depositate dal nastro stesso. Questi depositi, se non tempestivamente asportati, causano la maggior parte dei messaggi "LOAD ERROR" che vedrete apparire sul monitor. Quindi, ogni 10 o 20 ore di funzionamento del Datasette, è opportuno fare un po' di pulizia. (Diciamo che l'intervallo medio tra le operazioni di manutenzione non dovrebbe superare le 15 C90 usate, ma ciò dipende molto dalla qualità dei nastri di cui vi servite...) Munitevi quindi di una confezione di Cotton Floc o similari e di una boccetta di alcool denaturato. Imbevete il cotone con l'alcool, togliete l'eventuale cassetta dall'unità, premete EJECT e

REM-HW

PLAY e iniziate a strofinare delicatamente le due testine con il batuffolo di cotone. Lo vedrete facilmente diventare color marrone: sono tutti depositi di nastro che stavano sulle testine. Se il cotone si sporca troppo, prendete un altro bastoncino e ripetete l'operazione fino a che rimanga completamente bianco. A questo punto sarà opportuno dare una pulita anche al rullo di gomma. Poiché questo ruota in senso orario, ponetevi subito a destra del perno e premete il batuffolo imbevuto contro la gomma. Marrone, eh? Ripete l'operazione finché non rimanga bianco. Se volete fare le cose fino in fondo, munitevi di uno smagnetizzatore di testine (lo troverete nei negozi di prodotti per HI-FI) e passatelo più volte davanti alla testina nel modo suggerito dal manuale che troverete nella confezione. Prima

di usare una cassetta attendete che le testine si siano asciugate dai residui di alcool.

Una cosa che invece vi sconsiglio vivamente di fare con leggerezza è il riallineamento della testina di lettura/scrittura. Anche se il foro per il cacciavite sullo chassis del Datasette può essere allettante per una simile operazione, fatela con cautela. Se non possedete l'esperienza adeguata, saranno più i guai dei buoni risultati. Se continuate, malgrado la pulizia delle testine, a ricevere messaggi "LOAD ERROR" usando cassette non registrate da voi, solo allora significa che la vostra testina è fuori allineamento. Non resta che provare a ruotare la vitina dell'azimut di qualche grado a destra a sinistra fino a che non si carichino con successo tutti i programmi.

Quando il computer parla il linguaggio delle immagini

La computer grafica rappresenta un campo di applicazione dell'informatica relativamente nuovo, ma suscettibile di imprevedibili sviluppi. Questo volume, nato in collaborazione con alcune delle più specializzate istituzioni del settore, esamina tutte le possibilità di questa scienza nuova e affascinante: dall'animazione cinematografica e televisiva ai business graphics; dalla

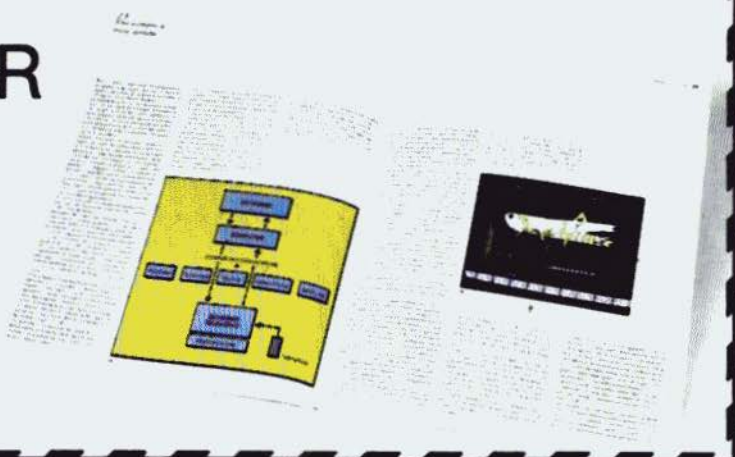
progettazione in architettura a quella in elettronica e in meccanica; dalla mappazione alla manipolazione tridimensionale delle immagini... Realizzata in modo da permettere un rapido, ma esauriente approccio all'argomento, l'opera si rivolge a quanti (lettori-utenti) siano alla ricerca dei necessari chiarimenti per una corretta e proficua utilizzazione delle tecniche di Computer grafica.

Mauro Salvemini

COMPUTER GRAFICA

176 pagine. Lire 29.000
Codice 519 P

GRUPPO EDITORIALE JACKSON



OK-List per la perfetta battitura dei listati

Inserite in memoria l'OK-LIST prima di accingervi a battere i listati pubblicati: avrete a disposizione un formidabile mezzo per essere sicuri di aver inserito nel computer ogni carattere in modo assolutamente corretto.

Tutti i nostri listati sono caratterizzati da un numero aggiunto ad ogni linea di programma, ad esempio: REM 123. **NON COPIATE QUESTA PARTE DELLA LINEA!** Serve unicamente per vostra informazione. L'istruzione REM ha l'unico scopo di renderla inoffensiva se per errore dovete batterla.

Se, prima di iniziare il lavoro di battitura del listato, caricate in memoria OK-LIST e lo attivate con RUN, per ogni linea che battere verrà visualizzato un numero (detto checksum) nell'angolo superiore sinistro dello schermo.

Confrontate questo numero con quello pubblicato per ogni linea del listato: se non corrisponde, ciò significa che avete commesso un errore nella battitura della linea stessa.

Le ore buttate per cercare e correggere errori di battitura in programmi che non funzionano saranno solo un lontano ricordo!

Due ultime cose di cui tenere conto: OK-LIST non tiene conto degli spazi: questo per vostra convenienza, poiché questi generalmente sono poco importanti.

Infine, OK-LIST è allocato nel buffer del registratore (locazioni 886-1018), quindi prima di salvare su cassetta il programma che state battendo, ricordatevi di disabilitarlo premendo contemporaneamente RUN/STOP + RESTORE. Potrete poi riattivarlo con SYS 886.

Il sistema più comodo per usare OK-LIST è quello di battere normalmente il programma, quindi listarlo e portarsi con il cursore sul numero di linea più basso che appare sul video; premere quindi RETURN e confrontare il numero di checksum visualizzato nell'angolo superiore sinistro del video con quello pubblicato per la data riga del programma. Il cursore si porterà automaticamente sulla linea successiva, e non dovrete fare altro che premere nuovamente RETURN per ottenere il checksum seguente.

Data l'area di memoria in cui si trova, OK-LIST non può essere usato per controllare un programma già salvato su cassetta, mentre ciò è possibile se il programma è stato registrato su disco.



OK-LIST versione per VIC 20 e C 64

```

100 PRINT"{CLR}ATTENDERE PREGO..."
105 FORI=886TO1018:READA:CK=CK+A:POKEI,A:
NEXT
110 IF CK<>17539 THEN PRINT"{GIU'}ERRORE
NELLE ISTRUZIONI DATA":END
120 SYS886:PRINT"{CLR}{ 2 GIU'}OK-LIST AT
TIVATO.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
  
```

MLX nuova versione

di C. Brannon
trad. e adatt. di F. Stella e
M. Cristulb Grizzi

Per l'introduzione senza errori di programmi in linguaggio macchina.

MLX è un programma che permette di inserire listati in linguaggio macchina in modo assolutamente sicuro da errori e senza la perdita di tempo del dover battere e controllare lunghe sequenze di istruzioni DATA.

Molti dei nostri listati di programmi in linguaggio macchina hanno il formato MLX (li riconoscete dal fatto che sono esclusivamente numerici), e richiedono quindi che MLX sia caricato in memoria ed eseguito prima della battitura del listato.

MLX vi chiederà l'indirizzo della locazione di partenza e l'indirizzo della lo-

cazione finale del programma da caricare. Questi valori sono sempre indicati nell'articolo che accompagna il listato.

MLX vi segnala automaticamente gli errori di battitura MENTRE STATE DIGITANDO IL LISTATO, e vi chiede di reinserire la linea errata. L'ultimo numero battuto di ogni linea rappresenta il checksum e viene visualizzato in reverse.

Sono inoltre disponibili altri comandi, quali:

SHIFT-N nuovo indirizzo: permette di cambiare l'indirizzo della linea che volete battere, ed è utile nel caso si inseriscano i listati di inizio e fine.

SHIFT-D display: lista i dati inseriti tra due indirizzi di inizio e fine.

SHIFT-L load: carica un file da nastro o disco in formato MLX.

SHIFT-S save: salva su nastro o disco un file in formato MLX.

Una volta battuto il listato e salvatolo tramite MLX, si potrà caricare direttamente il programma con un'istruzione LOAD "nome del file", 1,1 per il registratore, oppure LOAD "nome del file", 8,1 per l'unità a dischi. Il programma sarà mandato in esecuzione con una SYS (indirizzo di partenza).

Presentiamo in questo numero una versione migliorata di MLX per C64 pubblicato sullo scorso numero, unitamente ad una versione per VIC 20.

Chi avesse digitato la precedente versione può sostituire semplicemente le linee del programma che sono state modificate.

MLX versione per VIC 20

```
100 PRINT "{CLR}{PUR}";CHR$(142);CHR$(8);
                                         :rem 181
101 POKE788,194:REM DISABILITA RUN/STOP
                                         :rem 144
110 PRINT "{RVS}{ 14 SPAZI}"
                                         :rem 117
120 PRINT "{RVS}{DES}{OFF}{<*>}{RVS}
{DES}{DES}{ 2 SPAZI}{<*>}{OFF}{<*>}
{RVS}{RVS}"
                                         :rem 191
130 PRINT "{RVS}{DES}{<G>}{DES}
{ 2 DES}{OFF}{RVS}{<*>}{OFF}{<*>}
{RVS}{OFF}"
                                         :rem 122
140 PRINT "{RVS}{ 14 SPAZI}"
                                         :rem 120
200 PRINT "{ 2 GIU' } {PUR}{BLK} UN PROGRAMMA
PER L'IN-TRODUZIONE{ 2 SPAZI} DI ROUT
INEIN LINGUAGGIO MACCHINA";
205 PRINT "A PROVA DI ERRORE{ 5 GIU' }"
                                         :rem 226
210 PRINT "{BLK}{ 4 SU} INDIRIZZO DI PARTEN
ZA":INPUTS:F=1-F:CS=CHR$(31+119*F)
                                         :rem 52
220 IF<256ORS>32767THENGOSUB3000:GOTO210
                                         :rem 2
225 PRINT:PRINT:PRINT:PRINT
                                         :rem 123
230 PRINT "{BLK}{ 3 SU} INDIRIZZO CONCLUSIV
O":INPUTS:F=1-F:CS=CHR$(31+119*F)
                                         :rem 170
240 IF<256ORE>32767THENGOSUB3000:GOTO230
                                         :rem 234
250 IF<STHENPRINTCS;"{RVS} INDIRIZZO CONC
LUSIVO <INDIRIZZO DI PARTENZA
{ 3 GIU' }":GOSUB1000:GOTO230:rem 27
260 PRINT:PRINT:PRINT
                                         :rem 179
300 PRINT "{CLR}";CHR$(14):AD=S
                                         :rem 56
310 PRINTRIGHTS("0000"+MIDS(STR$(AD),2),5
);":":FORJ=1TO6
                                         :rem 234
```

```
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                         :rem 228
390 IFN=-211THEN710
                                         :rem 62
400 IFN=-204THEN790
                                         :rem 64
410 IFN=-206THENPRINT:PRINT "{GIU' } INSE
CI UN NUOVO{ 4 SPAZI} INDIRIZZO":INPU
TZZ
                                         :rem 42
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
{RVS}ESCE DAL CAMPO DI VA-"
                                         :rem 150
416 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
{RVS}LORI INDICATO":GOSUB1000:GOTO410
                                         :rem 105
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                         :rem 238
420 IFN<>-196THEN480
                                         :rem 133
430 PRINT:INPUT "LISTATO:DA";F:PRINT,"
{ 2 SIN}A";:INPUT
                                         :rem 29
440 IF<SORF>EORT<SORT>ETHENPRINT "MINIMO"
;S;"{SIN}, MASSIMO";E:GOTO430
                                         :rem 36
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHTS("000
0"+MIDS(STR$(I),2),5);":":
                                         :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
PC(10);
                                         :rem 34
457 PRINTRIGHTS("00"+MIDS(STR$(N),2),3);"
,";
                                         :rem 157
460 GETAS:IFAS>" THENPRINT:PRINT:GOTO310
                                         :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
T:GOTO310
                                         :rem 50
480 IFN<0THENPRINT:GOTO310
                                         :rem 168
490 A(J)=N:NEXTJ
                                         :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
SUM=(CKSUM+A(I))AND255:NEXT
                                         :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
```

```

:rem 234
515 IFN=CKSUMTHEN530 :rem 255
520 PRINT:PRINT"LA RIGA E' STATA INSE-
A IN MANIERA ERRATA" :rem 39
525 PRINT"RI-INSEIRILA":PRINT:GOSUB1000:G
OTO310 :rem 167
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT
:rem 80
550 AD=AD+6:IFAD<ETHEN310 :rem 212
560 GOTO710 :rem 108
570 N=0:Z=0 :rem 88
580 PRINT"<+>"; :rem 79
581 GETA$:IFA$=""THEN581 :rem 95
585 PRINTCHR$(20);A=ASC(A$):IFA=13ORA=44
ORA=32THEN670 :rem 229
590 IFA>128THENN=-A:RETURN :rem 137
600 IFA<>20THEN630 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
{SIN} {SIN}";GOTO690 :rem 172
620 GOTO570 :rem 109
630 IFA<48ORA>57THEN580 :rem 105
640 PRINTA$;N=N*10+A-48 :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
:rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570 :rem 114
680 PRINT";":RETURN :rem 240
690 S$=PEEK(209)+256*PEEK(210)+PEEK(211)
:rem 149
692 FORI=1TO3:T=PEEK(S-I) :rem 68
695 IFT<>44ANDT<>58THENPOKES-I,32:NEXT
:rem 205
700 PRINTLEFTS("{ 3 SIN}",I-1);RETURN
:rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***
{ 3 GIU'}" :rem 236
720 INPUT"{GIU'} NOME DEL FILE";F$
:rem 199
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)" :rem 128
740 GETA$:IFAS<>"N"ANDAS<>"D"THEN740
:rem 30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
:rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256 :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469 :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654

```

```

:rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
:POKE780,253 :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
:SYS65496 :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
:rem 111
775 PRINT"{GIU'}_OPERAZIONE CONCLUSA.":END
:rem 152
780 PRINT"{GIU'}ERRORE DI REGISTRAZIONE-
RIPROVA.":IFDV=1THEN720 :rem 8
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT#1$
;E2$:CLOSE15:GOTO720 :rem 103
782 GOTO720 :rem 115
790 PRINT"{CLR}{RVS}*** LOAD ***
{ 2 GIU'}" :rem 212
800 INPUT"{ 2 GIU'} NOME DEL FILE";F$
:rem 215
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)" :rem 127
820 GETA$:IFAS<>"N"ANDAS<>"D"THEN820
:rem 28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
:rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256 :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469 :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
66 :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
:rem 111
865 PRINT"{GIU'}_OPERAZIONE CONCLUSA.":GOT
O310 :rem 142
870 PRINT"{GIU'}ERRORE DI CARICAMENTO.RIP
ROVA.{GIU'}:IFDV=1THEN800 :rem 43
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT#1$
;E2$:CLOSE15:GOTO800 :rem 102
1000 REM CICALINO :rem 231
1001 POKE36878,15:POKE36874,190 :rem 206
1002 FORW=1TO300:NEXTW :rem 117
1003 POKE36878,0:POKE36874,0:RETURN
:rem 74
2000 REM CAMPANELLO :rem 130
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368
76,240:NEXTW :rem 22
2002 POKE36876,0:RETURN :rem 119
3000 PRINTC$;"{RVS}NON IN PAGINA ZERO O
{ 2 SPAZI}SU ROM{ 4 GIU'}":GOTO1000
:rem 23

```

MLX nuova versione per C64

```

100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);:
POKE53281,1:POKE53280,1 :rem 73
101 POKE788,52:REM DISABILITA RUN/STOP
:rem 89
110 PRINT"{RVS}{ 40 SPAZI}"; :rem 176
120 PRINT"{RVS}{ 15 SPAZI}{DES}{OFF}{<*>
{RVS}{DES}{DES}{ 2 SPAZI}{<*>}{OFF}
{<*>}{RVS}{RVS}{ 13 SPAZI}";
:rem 250
130 PRINT"{RVS}{ 15 SPAZI}{DES}{<N>}{<H>
{DES}{ 2 DES}{OFF}{RVS}{<*>}{OFF}
{<*>}{RVS}{ 13 SPAZI}"; :rem 220
140 PRINT"{RVS}{ 40 SPAZI}" :rem 120
200 PRINT"{ 2 GIU'}{PUR} UN PROGRAMMA
{ 3 SPAZI}PER{ 3 SPAZI}L'INTRODUZIONE
DI{ 2 SPAZI}ROUTINE IN LINGUAGGIO";

```

```

:rem 217
205 PRINT" MACCHINA A PROVA{ 17 SPAZI}DI
{ 2 SPAZI}ERRORE{ 3 GIU'}" :rem 244
210 PRINT"<5>{ 2 SU} INDIRIZZO DI PARTE
NZA{ 2 SPAZI}";:INPUTS:F=1-F:C$=CHR$(
31+119*F) :rem 85
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
THENGOSUB3000:GOTO210 :rem 235
225 PRINT:PRINT:PRINT :rem 180
230 PRINT"<5>{ 2 SU} INDIRIZZO CONCLUSI
VO{ 3 SPAZI}";:INPUTE:F=1-F:C$=CHR$(3
1+119*F) :rem 92
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
THENGOSUB3000:GOTO230 :rem 183
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
LUSIVO<INDIRIZZO INIZIALE" :rem 200

```

```

255 IFE<STHENGOSUB1000:GOTO230 :rem 119
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0 :rem 225
310 PRINTRIGHT$( "0000"+MID$(STR$(AD),2),5 :rem 234
);":":FORJ=1TO6
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320 :rem 228
390 IFN=-211THEN710 :rem 62
400 IFN=-204THEN790 :rem 64
410 IFN=-206THENPRINT:INPUT"{GIU'} INSERT :rem 13
RE IL NUOVO INDIRIZZO";ZZ
414 IFN=-206THENIFZZ<SORZZ>ETHENPRINT" :rem 105
[RV$]ESCE DAL CAMPO DI VALORI INDICAT
O"
415 IFN=-206THENIFZZ<SORZZ>ETHENGOSUB1000 :rem 202
:GOTO410
417 IFN=-206THENAD=ZZ:PRINT:GOTO310 :rem 238
420 IFN<>-196THEN480 :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT" :rem 183
{ 9 SPAZI}A";:INPUTT
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO" :rem 147
;S;" MASSIMO";E;"!(<5>":GOTO430
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$( "000 :rem 30
0"+MID$(STR$(I),2),5);":
451 FORK=OTO5:N=PEEK(I+K):PRINTRIGHT$( "00 :rem 66
"+MID$(STR$(N),2),3);":
460 GETA$:IFA$>" THENPRINT:PRINT:GOTO310 :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN :rem 50
T:GOTO310
480 IFN<0THENPRINT:GOTO310 :rem 168
490 A(J)=N:NEXTJ :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK :rem 200
SUM=(CKSUM+A(I))AND255:NEXT
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20) :rem 234
515 IFN=CKSUMTHEN530 :rem 255
520 PRINT:PRINT"{RED}LA LINEA E' STATA IN :rem 157
SERITA IN MANIERA"
525 PRINT"ERRATA. RIPETERE(<5>":PRINT:GO :rem 27
SUB1000:GOTO310
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54 :rem 227
272,0:POKE54273,0
550 AD=AD+6:IFAD<ETHEN310 :rem 212
560 GOTO710 :rem 108
570 N=0:Z=0 :rem 88
580 PRINT"(<+>"; :rem 79
581 GETA$:IFA$=" THEN581 :rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44 :rem 229
ORA=32THEN670
590 IFA>128THENN=-A:RETURN :rem 137
600 IFA<>20THEN630 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT" :rem 172
{SIN}{SIN}";:GOTO690
620 GOTO570 :rem 109
630 IFA<48ORA>57THEN580 :rem 105
640 PRINTA$;:N=N*10+A-48 :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600 :rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570 :rem 114
680 PRINT",";:RETURN :rem 240
690 S$=PEEK(209)+256*PEEK(210)+PEEK(211) :rem 149
691 FORI=1TO3:T=PEEK(S$-I) :rem 67
695 IFT<>44ANDT<>58THENPOKE$-I,32:NEXT :rem 205

```

```

700 PRINTLEFT$( "{ 3 SIN",I-1);:RETURN :rem 7
710 PRINT"{CLR}{RV$}*** SAVE *** :rem 236
{ 3 GIU' }"
720 INPUT"{GIU'}NOME DEL FILE";F$ :rem 199
730 PRINT:PRINT"{ 2 GIU'}{RV$}N{OFF}ASTRO :rem 128
O {RV$}D{OFF}ISCO: (N/D)"
740 GETA$:IFA$<>"N"ANDAS<>"D"THEN740 :rem 30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$ :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$ :rem 3
):POKE782,ZK/256
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN( :rem 109
T$):SYS65469
763 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 69
66
765 POKE254,S/256:POKE253,S-PEEK(254)*256 :rem 12
:POKE780,253
766 POKE782,E/256:POKE781,E-PEEK(782)*256 :rem 124
:SYS65496
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780 :rem 111
775 PRINT"{GIU'}FATTO.":END :rem 194
780 PRINT"{GIU'}ERRORE NEL SAVE-RIPROVA!" :rem 104
:IFDV=1THEN720
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$ :rem 103
;E2$:CLOSE15:GOTO720
790 PRINT"{CLR}{RV$}*** LOAD *** :rem 212
{ 2 GIU' }"
800 INPUT"{ 2 GIU'}NOME DEL FILE";F$ :rem 215
810 PRINT:PRINT"{ 2 GIU'}{RV$}N{OFF}ASTRO :rem 127
O {RV$}D{OFF}ISCO: (N/D)"
820 GETA$:IFA$<>"N"ANDAS<>"D"THEN820 :rem 28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$ :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$ :rem 2
):POKE782,ZK/256
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN( :rem 107
T$):SYS65469
845 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 70
66
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870 :rem 111
865 PRINT"{GIU'}FATTO.":GOTO310 :rem 184
870 PRINT"{GIU'}ERRORE NEL LOAD-RIPETI! :rem 19
{GIU'}":IFDV=1THEN800
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$ :rem 102
;E2$:CLOSE15:GOTO800
1000 REM CICALINO :rem 231
1001 POKE54296,15:POKE54277,45:POKE54278, :rem 207
165
1002 POKE54276,33:POKE54273,6:POKE54272,5 :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54 :rem 202
273,0:POKE54272,0:RETURN
2000 REM CAMPANELLO :rem 130
2001 POKE54296,15:POKE54277,0:POKE54278,2 :rem 152
47
2002 POKE54276,17:POKE54273,40:POKE54272, :rem 86
0
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN :rem 57
3000 PRINTC$;"{RV$} NON IN PAGINA ZERO O :rem 240
SU{DES}ROM ":GOTO1000

```



di T. Reynolds
trad. e adatt.
di G. G. Barzaghi

Mentre date la scalata alla parete di un palazzo, aggirando le finestre, dovete fare attenzione ai vasi di fiori che cadono ed all'attacco degli uccelli. Originariamente scritto per il VIC, ne viene fornita anche la versione per il C64. È necessario un joystick.

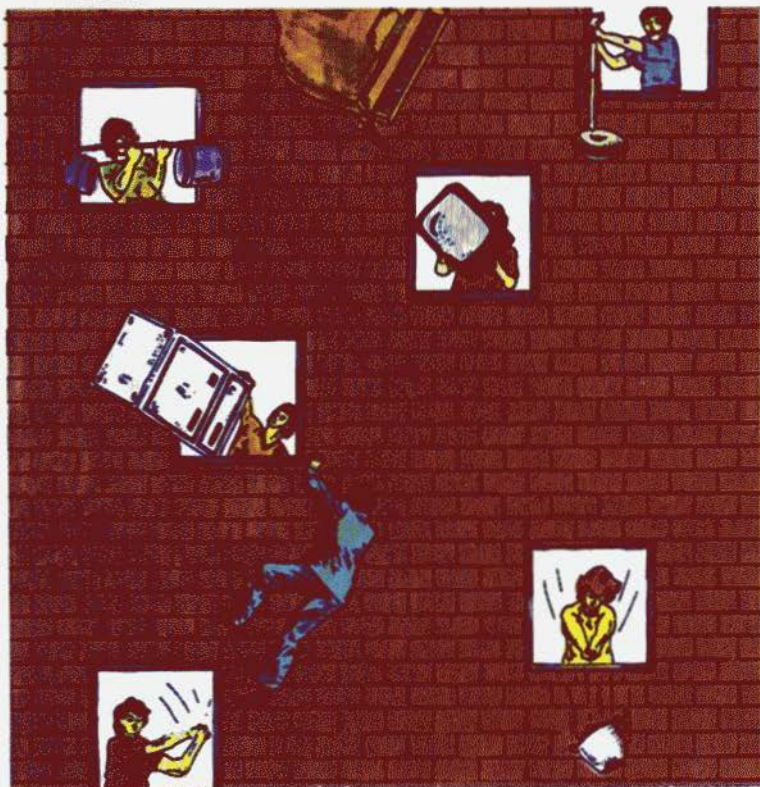
Lo scalatore folle è un gioco che richiede abilità manuale e capacità di giudizio. Con il joystick, controllate lo "scalatore folle" mentre sale lungo la facciata in mattoni di un palazzo, evitando le finestre e gli oggetti che cadono. Quanto più in alto sale, maggiore è il vostro punteggio.

La versione per VIC 20

Cominciate col copiare il programma 1 e salvatelo su nastro. Quindi battete il programma 2 e registratelo subito dopo il programma 1 sullo stesso nastro. Quando il programma 1 viene mandato in esecuzione, caricherà automaticamente il programma 2 da nastro e lo manderà in esecuzione.

Quando il gioco parte vedrete una dimostrazione dello scalatore folle in azione. Dopo questa breve dimostrazione, si fermerà e voi potrete partire premendo il pulsante di sparo.

Se lo "scalatore folle" cade, il gioco termina e viene mostrato il vostro punteggio. Limitatevi a premere il pulsante di sparo per dare il via ad un'altra manche. Lo scalatore cadrà se entrerà in contatto, in qualsiasi modo, con una finestra. La scalata è semplice fino a che raggiungete i 100 punti. Da questo momento, dovete verdevela con i vasi di fiori che cadono. Da 300 punti in poi, i vasi di fiori cesseranno di cadere, ma dovete guardarvi dagli uccelli che passano volando dal lato sinistro dello schermo. Se raggiungete i 600 punti, ce l'avete fatta a raggiungere la cima



del palazzo, e ricominciate alla base del palazzo successivo.

La versione per C64

In questa versione, ci sono alcune differenze fondamentali nel gioco. Innanzi tutto, inserite il joystick nella porta 1. Premete la barra spaziatrice per far partire il gioco. Contrariamente alla versione VIC, potete arrampicarvi lungo una finestra, fintanto che vi

manteniate in contatto con il muro. Verranno scagliati verso di voi una quantità di oggetti — televisori, pianoforti, casseforti (è uno di quei pazzeschi condomini) — e raddoppiano la loro velocità di caduta quando superate i 150 punti.

Questa versione registra inoltre i punteggi record. Premendo la barra spaziatrice si dà il via ad una nuova partita.

Lo scalatore folle versione per VIC 20

Programma 1

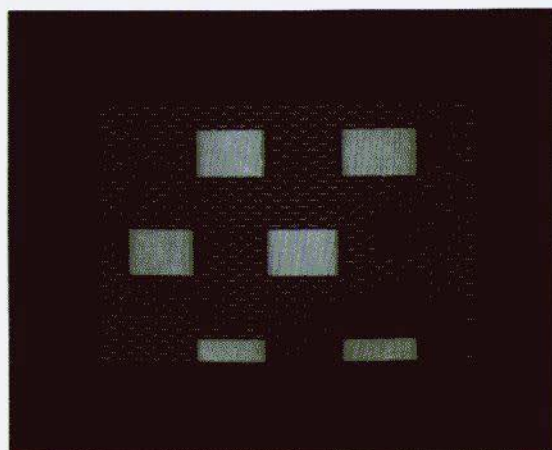
```
90 POKE56,28:POKE52,28:CLR          :rem 26
100 FORA=7168TO7432:READD:POKEA,D:NEXT :rem 108
```

```
110 PRINT"{CLR}{ 4 SPAZI}"CHR$(34)"SCALAT
    ORE #2"CHR$(34)                  :rem 159
115 POKE631,19:POKE632,131:POKE198,2:NEW
    :rem 72
120 DATA0,247,247,247,0,127,127,127,0,119
    ,119,119,0,127,127,127          :rem 69
130 DATA0,247,199,199,0,67,99,97,0,247,22
    7,193,0,65,65,99                :rem 81
```

```

140 DATA0,247,227,227,0,71,7,15,0,247,247
    ,247,0,127,113,113 :rem 118
150 DATA0,247,227,193,0,65,65,99,0,247,24
    7,247,0,127,15,15 :rem 94
160 DATA64,170,85,42,60,60,60,24,191,85,1
    70,213,195,195,195,231 :rem 91
170 DATA0,240,244,247,0,127,127,127,0,0,0
    ,0,0,0,0 :rem 193
180 DATA0,23,55,247,0,127,127,127,0,240,2
    40,247,0,127,127,127 :rem 216
190 DATA0,0,0,0,0,0,0,0,0,7,7,119,0,127,1
    27,127 :rem 255
200 DATA143,135,195,225,240,0,0,240,255,2
    55,255,255,31,95,15,31 :rem 65
210 DATA0,247,247,247,0,127,127,127,0,0,0
    ,23,0,31,31,31 :rem 151
220 DATA0,7,7,199,0,71,71,71,0,224,224,22
    4,0,99,99,99 :rem 105
230 DATA0,0,0,240,0,120,120,120,0,247,247
    ,247,0,127,127,127 :rem 87
240 DATA55,255,255,255,248,250,240,248,2
    41,225,195,135,15,0,0,15 :rem 177
250 DATA0,247,247,247,0,127,127,127,0,23,
    23,247,0,127,127,127 :rem 223
260 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    :rem 70
270 DATA0,240,240,247,0,127,127,127,255,2
    55,255,255,255,255,255 :rem 30
280 DATA0,0,0,0,0,0,0,0,0 :rem 196

```



Programma 2

```

1 BN=1:Z=1:GOTO8 :rem 220
2 PRINT"{HOME}{ 2 GIU'";BKS:ONZGOTO27,55
    :rem 141
3 PRINT"{HOME}{ 2 GIU'";WS:ONZGOTO27,55
    :rem 88
4 PRINT"{HOME}{ 2 GIU'";WLS:ONZGOTO27,55
    :rem 138
5 PRINTBKS:GOTO66 :rem 82
6 PRINTWS:GOTO66 :rem 29
7 PRINTWLS:GOTO66 :rem 79
8 FORA=828TO907:READD:POKEA,D:NEXTA
    :rem 240
9 DATA169,233,133,1,169,31,133,2,169,2,13
    3,0,162,242,160,0 :rem 69
10 DATA177,1,160,22,145,1,198,1,169
    :rem 228

```

```

11 DATA255,197,1,208,2,198,2,202,208,235,
    198,0,208,229,96 :rem 50
12 DATA169,233,133,1,169,151,133,2,169,2,
    133,0,162,242,160,0 :rem 162
13 DATA177,1,160,22,145,1,198,1,169
    :rem 231
14 DATA255,197,1,208,2,198,2,202,208,235,
    198,0,208,229,96 :rem 53
15 P1=37151:P2=37152:LC=7922 :rem 250
16 PRINT"{CLR}":POKE36869,255:POKE36879,8
    :X=6:POKE36876,220:POKE37154,127
    :rem 89
17 SN=36876:VL=36878 :rem 125
18 AS(1)="{RED}ECD{GIU'}{ 3 SIN}MKL{GIU'}
    { 2 SIN}ST{GIU'}{ 3 SIN}["AS(2)="
    {RED}BCG{GIU'}{ 3 SIN}JKO{GIU'}
    { 3 SIN}UV{GIU'}{SIN}t" :rem 19
19 WS="{RED}@@{CYN}----{RED}@@@{CYN}----
    {RED}@@@@@" :rem 48
20 BLS="{RED}@@{GIU'}{ 3 SIN}@@{GIU'}
    { 3 SIN}@@{GIU'}{ 3 SIN}@@{GIU'} :rem 77
21 MPS="{HOME}{ 11 GIU'"; :rem 115
22 BKS="{RED}@@@@@@@@@@@@@@@@@@@@@@@@
    :rem 242
23 WLS="{RED}@@@@@{CYN}----{RED}@@@@
    {CYN}----{RED}@@@" :rem 92
24 DNS="{ 22 GIU'"; :rem 19
25 PRINT"{GIU'";:FORA=0TO19:PRINTBKS:NEX
    T :rem 81
26 PRINT"{HOME}{ 2 GIU'":FORA=0TO3:PRINT
    WS:NEXT :rem 206
27 PRINTMPS;TAB(X+M);AS(1+M):FORA=0TO100:
    NEXT :rem 0
28 PRINTMPS;TAB(X+M);BLS:SYS828:SYS868:PR
    INT"{HOME}{RVS}";G :rem 68
29 POKE36878,5:POKE36878,0 :rem 224
30 IFM=0THENM=1:GOTO32 :rem 110
31 IFM=1THENM=0:GOTO32 :rem 111
32 G=G+1:IFG>=50ANDG<58THENPRINT"{HOME}
    { 2 GIU'}{RVS}{WHT}{ 3 SPAZI}{RED}C
    {CYN}A{PUR}L{GRN}A{WHT}T{BLU}O{YEL}R
    {WHT}E {RED}F{CYN}O{PUR}L{GRN}L{BLU}E"
    :GOTO27 :rem 71
33 IFG=58THEN36 :rem 126
34 L=L+1:ONLGOTO2,2,2,2,3,3,3,3,2,2,2,2
    :rem 52
35 L=0:GOTO34 :rem 251
36 PRINTMPS;TAB(X+M);AS(1+M) :rem 132
37 FORA=1TO1000:NEXT=Y=7 :rem 228
38 PRINT"{CLR}{GIU'":FORA=0TO19:PRINTBKS
    :NEXT :rem 173
39 PRINT"{HOME}{ 2 GIU'":FORA=0TO4:PRINT
    WLS:NEXT :rem 4
40 PRINT"{HOME}{ 22 GIU'";BKS;"{HOME}"
    :rem 60
41 PRINTMPS;LEFTS(DNS,Y);TAB(X+M);AS(1+M)
    :rem 150
42 IF(PEEK(37151)AND32)<>0THEN42 :rem 52
43 Z=2:G=0 :rem 30
44 J0=-((PEEK(P1)AND16)=0):J1=-((PEEK(P1)
    AND8)=0) :rem 188
45 J2=-((PEEK(P1)AND4)=0):J3=-((PEEK(P2)A
    ND128)=0) :rem 242
46 IFG>=100ANDG<299THEN67 :rem 226
47 IFG>=300ANDG<599THEN72 :rem 228
48 IFG>=600THEN77 :rem 240
49 IFPEEK(P1)+PEEK(P2)=373THEN44 :rem 135
50 PRINTMPS;LEFTS(DNS,Y);TAB(X+M);BLS;"
    {HOME}{RVS}";SC :rem 94
51 IFJ0THENX=X-1:XL=1:IFX<0THENX=0
    :rem 182

```



```

52 IFJ1THENG=G-1:SC=SC-1:YL=-1:Y=Y+1:IFY>
7THENY=7:G=G+1:SC=SC+1          :rem 188
53 IFJ2THENG=G+1:SC=SC+1:YL=1:Y=Y-1:IFY<0
THENY=0:GOTO59                    :rem 0
54 IFJ3THENX=X+1:XL=-1:IFX>17THENX=17
                                     :rem 89
55 M=(255-M)AND1:POKE36878,5:POKE36878,0
                                     :rem 14
56 ZX=LC+X+M+(22*Y)                :rem 71
57 IFPEEK(ZX)ORPEEK(ZX+2)ORPEEK(ZX+66)ORP
EEK(ZX+68)THEN63                  :rem 163
58 PRINTMPS;LEFTS(DNS,Y);TAB(X+M);AS(1+M)
:GOTO44                            :rem 121
59 SYS828:SYS868:IFPT=1THENPP=PP+22:PM=PM
+1                                  :rem 17
60 IFBD=1THENBP=BP+22               :rem 8
61 L=L+1:ONLGO2,2,2,2,3,3,3,3,2,2,2,4
,4,4,4                              :rem 180
62 L=0:GOTO61                       :rem 251
63 PRINT"[HOME]{" 22 GIU'"}";:POKESN,250:PO
KEVL,10:FORA=1TOG                  :rem 29
64 L=L+1:ONLGO6,6,6,6,5,5,5,5,7,7,7,5
,5,5,5                              :rem 231
65 POKESN,PEEK(SN)-1:L=0:GOTO64     :rem 172
66 NEXTA:PRINT"[CLR]":X=6:L=0:POKESN,220:
POKEVL,0:PRINT"[RVS]{WHT}PUNTI:"G"
{RED}":SC=SC-G:GOTO37             :rem 58
67 IFPT=1THENPM=PM+1:IFPM=>20THENPT=0:PM=
0:POKEPP,PL                        :rem 168
68 IFPT=0THENPP=INT(RND(1)*21)+7724:PT=1:
PL=PEEK(PP)                        :rem 189
69 POKEPP,PL:PP=PP+22:PL=PEEK(PP)
                                     :rem 117
70 IFPL<>31ANDPL<>0THENPL=0:PP=7746:GOTO6
3                                  :rem 140
71 POKEPP,9:GOTO49                  :rem 124
72 IFBD=1THENBH=BH+1:IFBH=>19THENBD=0:BH=
0:POKEBP,BL:POKEBP-1,B1           :rem 0
73 IFBD=0THENBP=INT(RND(1)*10)*22+7725:BD
=1:BL=PEEK(BP):B1=PEEK(BP-1)      :rem 48
74 POKEBP,BL:POKEBP-1,B1:BP=BP+1:BL=PEEK(
BP):B1=PEEK(BP-1)                 :rem 50
75 IFBL<>31ANDBL<>0THENBL=0:BP=7746:GOTO6
3                                  :rem 89
76 POKEBP,17:POKEBP-1,16:GOTO49     :rem 142
77 PRINT"[CLR]":POKE36869,240:BN=BN+1
                                     :rem 0
78 PRINT"C'E' L'HAI FATTA!!":PRINT"ORA RI
PROVA COL PALAZ-ZO NR.":MID$(STRS(BN),
2)                                  :rem 253
79 FORA=1TO2500:NEXT:POKE36869,255:GOTO37
                                     :rem 7

```

```

130 POKE54276,129:POKE54273,10:POKE54272,
100                                :rem 184
140 PORT=1TO200:NEXT:POKE54276,128:RETURN
                                     :rem 62
150 S=54272:FORE=STOS+28:POKEE,0:NEXT
                                     :rem 46
160 POKE54296,15:POKE54277,52:POKE54278,2
44                                  :rem 160
170 POKE54276,17:POKE54272,2:RETURN
                                     :rem 72
180 FORHG=200TO0STEP-1:POKE54273,HG:NEXT
                                     :rem 237
190 POKE53281,1:POKE646,0:SC=0:HS=0
                                     :rem 13
200 TIS="000000"                   :rem 244
210 PRINT"[CLR]"SPC(3)"STO INSERENDO I DA
TI PER GLI SPRITE"                 :rem 71
220 PRINTSPC(3)"E LE ROUTINE IN LINGUAGGI
O MACCHINA"                         :rem 124
230 PRINT"[ 14 GIU']{ 3 DES}PAZIENTATE PE
R CIRCA QUATTRO MINUTI"            :rem 239
240 POKE2,0:POKE191,48             :rem 239
250 K=43008:L=1024                  :rem 168
260 PORT=12288TO13312+4*LSTEP3:POKET,239:
POKET+1,239:POKET+2,250           :rem 61
270 PRINT"[HOME]";SPC(18)"{ 7 GIU'"};RIGH
TS(TIS,3):NEXT                     :rem 218
280 FORR=1TO24                      :rem 74
290 READ DA                          :rem 60
300 PORT=DA TO DA+5:POKET,32:POKET+40,32:
POKET+80,32:POKET+120,32          :rem 48
310 NEXT:PRINT"[HOME]";SPC(18)"{ 7 GIU'"}
";RIGHTS(TIS,3):NEXT              :rem 78
320 DATA12332,12344,12358,12698,12712,130
12,13024,13038                     :rem 214
330 DATA13332,13344,13358,13698,13712,140
12,14024,14038                     :rem 223
340 DATA14332,14344,14358,14698,14712,150
12,15024,15038                     :rem 232
350 D1=0:D2=128:GOSUB620:GOSUB950:D1=64:D
2=-128:GOSUB660:GOSUB1460:POKE50688,
1                                  :rem 157
360 PRINT"[CLR]";:SYS49152:POKE53248,160:
SYS49517                            :rem 235
370 V=53248:SC=0:POKEV+21,255:POKEV+1,215
                                     :rem 100
380 V=53248:POKEV+39,1:IFPEEK(700)=0THENS
1=13:S2=15:POKE53272,20          :rem 218
390 IFSC>50ANDPEEK(V+1)>99AND((PEEK(56321)
)AND15)=14)THENPOKEV+1,(PEEK(V+1)-.3)
                                     :rem 25
400 IF((PEEK(53278)AND1)OR(PEEK(53279)AND
1)=0)THENQ=1:GOTO560              :rem 39
410 POKE700,0                       :rem 183
420 IFSC>150THENPOKE50688,0        :rem 84
430 SYS49176                         :rem 161
440 IFPEEK(52993)<20THENPOKE2041,228+RND(
0)*4:POKEV+40,(PEEK(2041)-228)+4
                                     :rem 82
450 POKEV+23,(PEEK(V+23)OR2):POKEV+29,(PE
EK(V+29)OR2)                       :rem 97
460 TR=ER:ER=RND(0)*320+24:IFPEEK(52993)<
230THENER=TR                       :rem 156
470 IFER>255THENIFER-255<30THENPOKEV+16,(
PEEK(V+16)OR2):POKEV+2,ER-255
                                     :rem 244
480 IFER<=255THENPOKEV+2,ER:POKEV+16,(PE
EK(V+16)AND253)                   :rem 68
490 IF(PEEK(56321)AND15)<>15THENPOKE2040,
S1:POKE54296,5:POKE54296,0       :rem 132
500 IF((PEEK(53278)AND1)OR(PEEK(53279)AND
1)=0)THENQ=1:GOTO560              :rem 40

```

Lo scalatore folle versione per C64

```

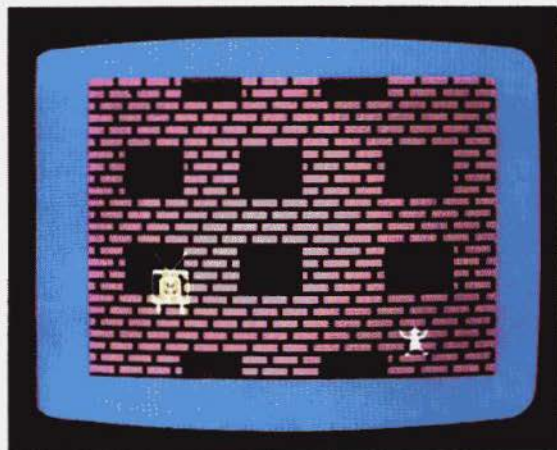
100 GOTO190                        :rem 100
110 S=54272:FORE=STOS+28:POKEE,0:NEXT
                                     :rem 42
120 POKE54296,15:POKE54277,51:POKE54278,2
52                                  :rem 154

```

```

510 POKEV+29,(PEEK(V+29)OR2):POKEV+23,(PE
EK(V+23)OR2) :rem 94
520 IF(PEEK(56321)AND15)=15THENPOKE2040,1
1 :rem 124
530 IFPEEK(700)THENS1=14:S2=11 :rem 231
540 IF((PEEK(53278)AND1)OR(PEEK(53279)AND
1)=0)THENQ=1:GOTO560 :rem 44
550 Q=0:GOTO590 :rem 105
560 POKEV+21,PEEK(V+21)AND1 :rem 56
570 GOSUB150:FORI=1TO100:SYS49469:POKEV+1
,(PEEK(V+1)+1)AND230 :rem 150
580 POKE54273,ABS(7-100):NEXT:Q=0:POKE532
78,0:GOSUB110:GOSUB1360 :rem 69
590 POKEV+21,255:SYS49176 :rem 189
600 IF(PEEK(56321)AND15)<>15THENPOKE2040,
S2:SC=SC+1 :rem 218
610 GOTO380 :rem 107
620 I1=832 :rem 236
630 READA:IFA=256THEN660 :rem 165
640 PRINT"{HOME}";SPC(18)"{ 7 GIU'";RIGH
TS(TIS,3) :rem 98
650 POKEI1,A:I1=I1+1:GOTO630 :rem 132
660 FORI=0TO20 :rem 62
670 FORJ=2TO0STEP-1 :rem 170
680 PRINT"{HOME}";SPC(18)"{ 7 GIU'";RIGH
TS(TIS,3) :rem 102
690 M1=832+D1+I*3+J:M2=832+D2+I*3+(2-J)
:rem 7
700 FORK=7TO0STEP-1 :rem 170
710 PRINT"{HOME}";SPC(18)"{ 7 GIU'";RIGH
TS(TIS,3) :rem 96
720 B1=2+K:B2=2+(7-K) :rem 159
730 V=-((PEEK(M1)ANDB1)<>0) :rem 224
740 POKEM2,PEEK(M2)OR(B2*V) :rem 80
750 NEXTK,J,I:RETURN :rem 43
760 DATA0,0,0,0,56,0,0 :rem 74
770 DATA124,6,0,254,6,0,254 :rem 89
780 DATA12,0,124,12,0,56,24 :rem 79
790 DATA7,255,240,15,255,224,24 :rem 43
800 DATA254,0,96,254,0,192,254 :rem 246
810 DATA0,0,254,0,0,254,0 :rem 225
820 DATA0,255,224,1,255,240,3 :rem 182
830 DATA0,24,6,0,14,12,0 :rem 177
840 DATA14,120,0,0,120,0,0 :rem 9
850 DATA122,0,0,0,0,56,0 :rem 175
860 DATA48,124,6,48,254,6,24 :rem 156
870 DATA254,12,24,124,12,12,56 :rem 237
880 DATA24,15,255,240,7,255,224 :rem 43
890 DATA0,254,0,0,254,0,0 :rem 233
900 DATA254,0,0,254,0,0,254 :rem 76
910 DATA0,0,254,0,1,254,0 :rem 227
920 DATA3,6,0,6,3,0,124 :rem 134
930 DATA1,224,120,1,224,0,0 :rem 67
940 DATA0,122,0,256 :rem 205
950 I=14592 :rem 41
960 READA:IFA=256THENRETURN :rem 239
970 PRINT"{HOME}";SPC(18)"{ 7 GIU'";RIGH
TS(TIS,3) :rem 104
980 POKEI,A:I=I+1:GOTO960 :rem 253
990 DATA0,0,0,0,6,0,0 :rem 26
1000 DATA28,0,0,120,0,0,252 :rem 57
1010 DATA0,3,230,0,15,131,0 :rem 55
1020 DATA62,1,128,127,255,248,127 :rem 135
1030 DATA255,252,85,85,92,127,255 :rem 150
1040 DATA252,68,0,20,68,0,20 :rem 128
1050 DATA64,0,16,0,0,0,0 :rem 169
1060 DATA0,0,0,0,0,0,0 :rem 57
1070 DATA0,0,0,0,0,0,0 :rem 58
1080 DATA0,255,255,224,191,255,160 :rem 185

```



```

1090 DATA192,0,96,223,255,96,223:rem 96
1100 DATA255,96,223,255,108,223,255 :rem 238
1110 DATA124,223,31,248,222,15,224 :rem 169
1120 DATA222,15,96,223,31,96,223:rem 82
1130 DATA255,96,223,255,96,223,255 :rem 199
1140 DATA96,223,255,96,223,255,96 :rem 155
1150 DATA128,0,32,255,255,224,224 :rem 129
1160 DATA0,224,224,0,224,224,0 :rem 218
1170 DATA224,255,0,0,0,0,0 :rem 15
1180 DATA0,0,0,0,0,0,0 :rem 60
1190 DATA0,0,0,0,0,0,16 :rem 116
1200 DATA0,8,56,0,28,56,0 :rem 237
1210 DATA28,255,255,255,255,255,255 :rem 248
1220 DATA56,0,28,56,0,28,16 :rem 88
1230 DATA0,8,0,0,0,0,0 :rem 64
1240 DATA0,0,0,0,0,0,0 :rem 57
1250 DATA0,0,0,0,0,0,0 :rem 58
1260 DATA0,0,0,0,0,0,4 :rem 63
1270 DATA0,32,2,0,64,1,0 :rem 174
1280 DATA128,0,129,0,0,66,0 :rem 80
1290 DATA0,36,0,15,255,240,8 :rem 135
1300 DATA0,80,8,255,16,9,255 :rem 142
1310 DATA144,9,153,144,9,231,144:rem 83
1320 DATA9,189,144,9,195,144,8 :rem 5
1330 DATA255,16,8,0,16,15,255 :rem 189
1340 DATA240,3,0,192,3,0,192 :rem 126
1350 DATA3,0,192,0,159,256 :rem 42
1360 POKEV+21,0 :rem 57
1370 POKE53281,1:PRINTCHRS(147);"{BLK}
{ 7 GIU'">{ 5 DES}SEI CADUTO" :rem 199
1380 PRINT"{ 5 DES">{ 3 GIU'"}IL TUO PUNTEG
GIO E' DI ";SC:IFSC>HSTHENHS=SC :rem 63
1390 PRINT"{ 5 DES">{ 3 GIU'"}IL PUNTEGGIO
RECORD E' ";HS :rem 170
1400 PRINT"{GIU'">{ 3 DES}PREMI IL PULSANT
E PER CONTINUARE" :rem 91
1410 FORI=1TO1000:GETAS:POKEI98,0:NEXT
:rem 50
1420 POKE2,0:POKEI91,48 :rem 32
1430 IFPEEK(V+3)>30THEN1430 :rem 25
1440 IFPEEK(197)=64ANDPEEK(V+3)>30THEN144
0 :rem 172

```

```

1450 POKE50688,1:SYS49152:POKEV+21,255:GO
      TO370 :rem 193
1460 I=49152 :rem 86
1470 READA:IFA=256THENRETURN :rem 28
1480 PRINT"[HOME]";SPC(18)"{ 7 GIU'";RIG
      HT$(TIS,3) :rem 149
1490 POKEI,A:I=I+1:GOTO1470 :rem 87
1500 DATA169,48,133,254,169,0,133
      :rem 141
1510 DATA253,141,252,207,141,253,207
      :rem 20
1520 DATA165,253,141,254,207,165,254
      :rem 33
1530 DATA141,255,207,173,254,207,133
      :rem 28
1540 DATA253,173,255,207,133,254,160
      :rem 31
1550 DATA0,177,253,153,0,4,200 :rem 229
1560 DATA208,248,230,254,160,0,177
      :rem 188
1570 DATA253,153,0,5,200,208,248:rem 81
1580 DATA230,254,160,0,177,253,153
      :rem 185
1590 DATA0,6,200,208,248,230,254:rem 81
1600 DATA160,0,177,253,153,0,7 :rem 233
1610 DATA200,192,232,208,246,173,1
      :rem 177
1620 DATA220,41,15,201,13,208,44:rem 67
1630 DATA24,169,40,109,254,207,141
      :rem 186
1640 DATA254,207,169,0,109,255,207
      :rem 192
1650 DATA141,255,207,56,173,254,207
      :rem 243
1660 DATA233,0,133,2,173,255,207:rem 80
1670 DATA233,52,5,2,144,10,169 :rem 240
1680 DATA48,141,255,207,169,0,141
      :rem 142
1690 DATA254,207,173,1,220,41,15:rem 81
1700 DATA201,14,208,51,56,173,254
      :rem 131
1710 DATA207,233,40,141,254,207,173
      :rem 229
1720 DATA255,207,233,0,141,255,207
      :rem 180
1730 DATA56,173,254,207,233,0,133
      :rem 134
1740 DATA2,173,255,207,233,48,5 :rem 41
1750 DATA2,176,17,24,173,254,207:rem 92
1760 DATA105,232,141,254,207,173,255
      :rem 30
1770 DATA207,105,3,141,255,207,169
      :rem 190
1780 DATA216,133,252,169,0,133,251
      :rem 186
1790 DATA169,10,160,0,162,4,145 :rem 34
1800 DATA251,200,208,251,230,252,202
      :rem 9
1810 DATA208,246,169,0,141,33,208
      :rem 136
1820 DATA173,1,220,41,15,201,7 :rem 226
1830 DATA208,36,173,16,208,41,1 :rem 35
1840 DATA240,7,173,0,208,201,60 :rem 26
1850 DATA176,22,24,173,0,208,105:rem 83
1860 DATA16,141,0,208,173,16,208:rem 84
1870 DATA105,0,141,16,208,169,1 :rem 32
1880 DATA141,188,2,173,1,220,41 :rem 33
1890 DATA15,201,11,208,36,173,16:rem 84
1900 DATA208,41,1,208,7,173,0 :rem 184
1910 DATA208,201,32,144,22,56,173
      :rem 130
1920 DATA0,208,233,16,141,0,208 :rem 23
1930 DATA173,16,208,233,0,141,16:rem 80
1940 DATA208,169,1,141,188,2,96 :rem 51
1950 DATA76,24,192,24,169,40,109:rem 100
1960 DATA254,207,141,254,207,169,0
      :rem 192
1970 DATA109,255,207,141,255,207,56
      :rem 248
1980 DATA173,254,207,233,0,133,2:rem 84
1990 DATA173,255,207,233,52,5,2 :rem 43
2000 DATA144,10,169,48,141,255,207
      :rem 181
2010 DATA169,0,141,254,207,76,24:rem 82
2020 DATA192,120,169,122,141,20,3
      :rem 117
2030 DATA169,193,141,21,3,88,96 :rem 47
2040 DATA165,161,166,161,142,0,207
      :rem 177
2050 DATA165,161,205,0,207,240,5:rem 72
2060 DATA169,0,141,1,207,238,1 :rem 232
2070 DATA207,173,1,207,141,3,208:rem 76
2080 DATA173,0,198,208,9,238,1 :rem 250
2090 DATA207,173,1,207,141,3,208:rem 78
2100 DATA76,49,234,256 :rem 113

```

**Aspettiamo i tuoi
lavori migliori**



di J. R. Krauss
trad. e adatt. di
M. Cristulib Grizzi

I programmatori BASIC possono "pokare" dati nella memoria "nascosta" del C64, ma per rileggere questi dati occorre operare la selezione a blocchi tra ROM e RAM. Il programma in linguaggio macchina qui presentato rende facile fare ciò che il BASIC non può ottenere direttamente, facendovi un regalino di 20 Kbyte extra.

Il Commodore 64 contiene 24 Kbyte di RAM (Random Access Memory) che non possono essere usati dal BASIC in modo diretto. Questa memoria è comunque accessibile dal chip VIC-II, così appare come il luogo ideale dove sistemare la pagina grafica oppure altri grossi insiemi di dati. Il guaio è che, benché sia possibile muovere dati in quest'area tramite istruzioni POKE, non è possibile rileggerli direttamente usando il BASIC. In questo articolo descriviamo una tecnica particolare che rende disponibile al programmatore BASIC tutta questa estesa porzione di memoria altrimenti non sfruttata. I microprocessori di molti piccoli computer possiedono al massimo 16 differenti linee di indirizzamento, cosicché questi computer possono indirizzare $2^{16} = 64$ Kbyte di RAM in modo diretto. Le macchine più costose sembrano possedere un più vasto ammontare di memoria indirizzabile, dal momento che la circuiteria preposta alla gestione della memoria ed il loro sistema operativo permettono di effettuare una selezione a banchi. Questa consiste nella possibilità di ottenere nelle stesse locazioni di memoria, a seconda delle necessità ed alternativamente, memoria ROM o RAM accessibile dal microprocessore. L'economico C64 non possiede una apposita unità di organizzazione della memoria, tuttavia può indirizzare 20 Kbyte di ROM con numerosi chip di Input/Output più 64 Kbyte di RAM. È un po' come riempire una bottiglia da un litro con quindici decilitri d'acqua... Ciò è possibile perché il microprocessore può selezionare il tipo di memoria (ROM o RAM) in vari blocchi, anche se posseggono gli stessi indirizzi. Nella sua configurazione normale, i

primi 2Kbyte della memoria del C64 sono usati come area di lavoro del sistema operativo e per la memoria di schermo. Della rimanente RAM, le locazioni da 2048 e 40959 sono disponibili per la programmazione in BASIC.

la corrispondente ROM. Per avere accesso alla RAM corrispondente, bisogna settare dei puntatori in modo che il microprocessore ignori la ROM: questi puntatori sono i bit 0, 1 e 2 della locazione 1. Le loro funzioni sono le se-

BIT	VALORE	SIGNIFICATO
0	1	Indica la normale BASIC ROM
	0	Indica la RAM non contigua (locaz. 40960-49151)
1	1	Indica la ROM del Kernal
	0	Indica la sottostante RAM (locaz. 57344-65535)
2	1	Indica i chip di I/O
	0	Indica il gen. di caratteri in RAM (locaz. 53248-57343)

Dopo questa locazione si trova l'interprete BASIC su ROM (40960-49151), mentre lo spazio oltre 40959 contiene 4 Kbyte di RAM (indirizzi 49152-53247), non contigui con l'area BASIC ed accessibili da BASIC solo tramite istruzioni PEEK e POKE. Seguono poi i dispositivi di I/O e la RAM colore (53248-57343), ed il Sistema Operativo Kernal (57344-65535). Ci sono altri 20 Kbyte di RAM così indirizzati: per sfruttarli occorre escludere la corrispondente ROM e quindi dire al processore di "vedere" la RAM invece della ROM. I registri interessati a questa operazione sono nelle locazioni 0 ed 1. Benché questa extra-RAM sia direttamente accessibile dal BASIC, i dati possono essere immagazzinati solo tramite PEEK e POKE. Se non si è prima operata la selezione del blocco di memoria, una PEEK in queste locazioni fornirà il valore immagazzinato nel-

guenti:

Se desideriamo immagazzinare una variabile - chiamiamola A - nella locazione, poniamo, 45000, e poi recuperarla in seguito, potremo scrivere una routine come questa:
10 POKE 45000,A: REM IMMAGAZZINA IL VALORE
20 POKE 1,54: REM SETTA IL PUNTATORE ALLA RAM
30 A=PEEK(45000): REM LEGGE IL VALORE
40 POKE 1,55: REM SETTA IL PUNTATORE ALLA ROM ORIGINARIA
50 END
Il risultato sarà un simpatico crash del computer. È facile vedere perché: la linea 20 dice al microprocessore di non "vedere" l'interprete BASIC in ROM, ma di considerarne tutte le corrispondenti locazioni come RAM. Così quando la linea 30 usa l'istruzione BASIC PEEK, essendo stato disabilitato l'in-



terprete BASIC, il computer non sa come interpretarla e si blocca. Se modificassimo il puntatore alla ROM Kernal o all'I/O, otterremmo ugualmente un blocco del sistema.

Dal momento che non possiamo ottenere un buon risultato tramite il BASIC, proveremo con il linguaggio macchina. Un caricatore BASIC implementerà la routine in l/m che si occupa di risolvere il nostro problema. Possiamo ora immagazzinare dati nella formalmente non disponibile RAM "mascherata" sotto la ROM.

La routine in linguaggio macchina, lunga solo 14byte, setta l'appropriato puntatore (il bit 0 nell'esempio) per ignorare la ROM (nell'esempio, l'interprete BASIC). Poi mette il byte di dati in una locazione normalmente accessibile dal BASIC (dal momento che la locazione 251 è inutilizzata in pagina zero, abbiamo scelto questa), e resetta il puntatore. (Un disassemblato di questa routine è mostrato alla fine dell'articolo).

Le linee 100-120, 340 e 350 non fanno realmente parte della routine - le linee 100 e 110 definiscono la variabile e la locazione di memoria da usare nell'esempio, e la linea 340 stampa il dato che la routine in linguaggio macchina ha immagazzinato per noi, per dimostrare che ha effettivamente funzionato. I 4 Kbyte di RAM che iniziano alla locazione 49152 non sono sfruttati, e dal momento che non sono contigui alla normale area BASIC, non possono disturbare eventuali programmi BASIC; così abbiamo scelto questa zona per immagazzinare la nostra routine. La linea 130 pone la locazione 49152 come locazione di partenza della routine.

Le linee 140, 160, 170 e 180 determi-

nano il valore che deve essere contenuto nella locazione 1, e mettono questo valore nella variabile BL. I numeri nelle istruzioni DATA (linee 190-200), sono i bytes del codice macchina, in notazione decimale. Tre di questi hanno valore zero. Il primo conterrà il puntatore al blocco BL. Il secondo ed il terzo, rispettivamente, conterranno il byte meno significativo e quello più significativo dell'indirizzo della locazione scelta nella RAM "nascosta". Le linee 210-240 caricano in memoria la routine, mentre la 250 inserisce il valore del puntatore al blocco BL nella routine stessa. Le linee 260 e 270 calcolano il byte meno e più significativo rispettivamente dell'indirizzo scelto, e le 280 e 290 li inseriscono al loro posto nella routine. La linea 300 disabilita le interruzioni in modo che la tastiera non venga letta durante l'esecuzione della routine in linguaggio macchina: ciò elimina la possibilità che l'interrupt venga generato nel momento non opportuno bloccando il computer.

L'apparentemente insignificante linea 310 provoca la vera e propria azione del programma, mandando in esecuzione la routine in codice macchina che, una volta compiute le operazioni, restituisce il controllo al BASIC. Ora possiamo recuperare il nostro byte di dati dove meglio ci aggrada (nello

esempio appunto nella locazione 251).

Si noti che la routine in linguaggio macchina può virtualmente risiedere in qualsiasi zona della memoria, anche nella RAM "nascosta". Se caricata nella normale area BASIC, ovviamente occorrerà modificare i puntatori appropriati del BASIC per proteggerla da sovrascritture. Dalla linea 140 in poi, la routine è perfettamente generica e può essere usata per leggere valori immagazzinati in qualsiasi locazione RAM nell'intervallo 0-65535, con l'eccezione di quella sotto l'area di Input/Output (53248-57343). Per vedere i 4 Kbyte di RAM sottostanti questa ROM è necessaria un'altra tecnica, in quanto in quest'area ci sono tre "strati" di memoria, e la nostra routine non ne copre il secondo dandoci accesso diretto al set di caratteri su ROM. Ciò può essere molto utile in programmi che usino caratteri definiti dall'utente, per riabilitare selettivamente porzioni della tavola di caratteri ROM.

Infine, bisogna notare due cose: questa routine può essere usata anche per leggere locazioni di memoria in cui risiedono sia il programma BASIC che la stessa routine, ma non si possono logicamente fare delle POKE indiscriminate in queste locazioni in quanto modificherebbero il programma.

DISASSEMBLATO DELLA ROUTINE

LDX	#500	:puntatore al blocco nel registro X
STX	\$01	:carica nella locazione 1
LDX	\$00	:contenuto della RAM nascosta nel registro X
STX	\$FB	:carica nella locazione 251 (decimale)
LDX	\$37	:valore normale del puntatore (55 decimale)
STX	\$01	:riporta al normale contenuto la locazione 1
RTS		:ripassa il controllo al BASIC

```

100 A=3:REM METTE NELLA VARIABILE "A" IL
    BYTE DI DATI DESIDERATO :rem 187
110 AD=45000:REM MEMORIZZEREMO "A" NELLA
    LOCAZIONE 45000 (RAM NASCOSTA) :rem 123
120 POKEAD,A:REM MEMORIZZA "A" :rem 21
130 MS=49152:REM IL CODICE MACCHINA SARA'
    CARICATO A PARTIRE DA 49152:rem 122
140 IF40959<ADANDAD<49152THENBL=54:GOTO19
    0 :rem 249
150 REM LA LOCAZIONE 1 CONTERRA' IL PUNTA
    TORE AL BLOCCO BL :rem 133
160 IF53247<ADANDAD<57344THENBL=51:GOTO19
    0 :rem 244
165 REM BL=51 -- ROUTINES DI I/O DISATTIV
    ATE :rem 26
170 IF57343<ADANDAD<65535THENBL=53:GOTO1
    90 :rem 54
175 REM BL=53 -- ROM KERNAL DISATTIVATA
    :rem 247
180 BL=55:REM NELLA NORMALE AREA BASIC
    :rem 225
190 DATA162,00,134,1:REM ROUTINE IN CODIC
    E MACCHINA :rem 176
200 DATA174,000,0000,134,251,162,55,134,1
    ,96 :rem 143
210 FORI=0TO13:REM CICLO PER IL CARICATOR
    E BASIC :rem 122

```

```

220 READML:REM LEGGE IL PROSSIMO BYTE DEL
    LA ROUTINE IN CODICE MACCHINA :rem 10
230 POKEMS+I,ML:REM CARICA IL BYTE IN MEM
    ORIA :rem 200
240 NEXT :rem 213
250 POKEMS+1,BL:REM METTE IL PUNTATORE AL
    BLOCCO NEL SECONDO BYTE DELLA ROUTIN
    E :rem 85
260 HA=INT(AD/256):REM BYTE ALTO DELL'IND
    IRIZZO DELLA RAM NASCOSTA :rem 21
270 LA=AD-256*HA:REM BYTE BASSO DELL'INDI
    RIZZO DELLA RAM NASCOSTA :rem 215
280 POKEMS+5,LA:REM CARICA GLI INDIRIZZI
    NELLA ROUTINE IN CODICE MACCHINA :rem 223
290 POKEMS+6,HA :rem 128
300 POKE56333,127:REM DISABILITA LE INTER
    RUZIONI :rem 195
310 SYS(MS):REM ESEGUE LA ROUTINE IN CODI
    CE MACCHINA :rem 133
320 POKE56333,129:REM RIABILITA LE INTERRU
    ZIONI :rem 130
330 A=PEEK(251):REM LEGGE IL BYTE DI DATI
    :rem 28
340 PRINTA:REM E LO STAMPA :rem 41
350 END :rem 111

```



di C. Dan
trad e adatt. di
R. Comini

Uno spettacolare viaggio all'interno del VIC 20. I signori viaggiatori sono pregati di allacciarsi le cinture, sta per cominciare il giro turistico attraverso la memoria del VIC.

Il programma è fondamentalmente scritto in codice macchina: questo prende dal VIC due pagine di memoria alla volta (ogni pagina è costituita da 256 byte) e le visualizza sullo schermo sotto forma di caratteri.

Questo algoritmo permette la visione di ciò che si vuole: ci si potrà muovere a piacimento all'interno della memoria, sia in avanti che indietro.

Se si volesse invece vedere una specifica area di memoria c'è una opzione che consente di saltare subito all'indirizzo desiderato. Il programma altro non è che un loader che da BASIC carica in memoria, tramite le istruzioni READ, DATA e POKE, un algoritmo in codice macchina. Come è noto un errore nella trascrizione dei numeri corrispondenti al codice macchina potrebbe causare un "blocco" al VIC, per cui è utile, prima di dare il RUN, avere il programma salvato su disco o cassetta. Questo sistema eviterà in caso di un banale errore, la nuova battitura di tutto il programma. L'uso del linguaggio macchina è una scelta dovuta alla sua elevata velocità. In poche parole ciò che si fa è leggere 512 byte di memoria da un qualunque punto di questa e trasferirli nella memoria di schermo, rendendoli così visibili sul monitor.

Poiché queste righe non sono dirette ad esperti conoscitori del linguaggio macchina, non entreranno in dettagli riguardo al funzionamento dell'algoritmo. Tuttavia i più esperti sappiano che la tecnica qui usata è assai interessante. Il programma descrive se stesso mentre gira. In poche parole lo si vede girare in modo reale.

Uso del programma

Per usare correttamente il programma, spegnere il VIC e togliere qualunque espansione o cartuccia, dopodiché riaccenderlo. Dopo aver battuto e sal-

vato il tutto dare il RUN. Primariamente si avrà una pausa di circa 2 secondi (tempo in cui viene "pokato" in memoria il linguaggio macchina), poi lo schermo si riempirà istantaneamente di caratteri. Quello che appare sullo schermo sono le prime due pagine di memoria del VIC, dall'indirizzo 0 al 505. Il programma si controlla con gli speciali tasti funzione e con il tasto —. Ecco un rapido riassunto delle funzioni svolte da detti tasti.

F1

Premendo il tasto F1 si percorre in avanti la memoria. Ogni volta che F1 è premuto si avanza di 40 byte (circa due linee di schermo). Tenendo premuto il tasto si ha un avanzamento continuo e rapido.

F3

Premendo F3 si percorre la memoria all'indietro. Anche qui per ogni pressione si ha uno scorrimento di 40 byte. Tenendo premuto F3 si ha uno scorrimento continuo a ritroso.

F5

Premendo F5 viene visualizzato l'indirizzo di inizio dei 506 byte presenti sullo schermo. Si vedrà un numero rosso lampeggiante nell'angolo in alto a sinistra: questo è l'indirizzo del primo byte visibile (è il primo carattere dello schermo). Poiché lo schermo ha 506 locazioni, l'ultimo byte visibile (nell'angolo in basso a destra) avrà come indirizzo quello del primo byte + 505. I caratteri che appaiono sullo schermo sono il risultato dato dai valori (codici schermo) dei byte che si stanno esaminando. Ad esempio, se un byte contiene il valore zero verrà mostrato sullo schermo sotto forma di chio-cioletta.

F7

Premendo F7 si ha un ritorno all'indirizzo di memoria ZERO. Si torna così all'inizio e sullo schermo appare la memoria dall'indirizzo 0 al 505.

Premendo — si entra in stato di input e compare il cursore lampeggiante: viene a questo punto richiesto l'indirizzo desiderato. Inserendo a questo punto un indirizzo a piacere (purché compreso tra 0 e 65275) e premendo il RETURN verranno visualizzati 506 byte a partire dall'indirizzo scelto.

Comincia il viaggio

Una volta lanciato il programma si preme F7, sono così visibili i primi 506 byte di memoria. Nella metà superiore dello schermo, leggermente sulla sinistra di questo sono il jiffy clock, ovvero l'orologio interno del VIC che sta contando il tempo. Si provi ora a premere un qualunque tasto (esclusi ovviamente i tasti di controllo), e due byte cambieranno. Questi sono i due byte che rilevano la pressione di un tasto e la segnalano al sistema operativo. Il valore di questi due byte (sono il byte 197 ed il 203) riflette precisamente il codice del tasto premuto. Sul margine destro si possono osservare ben 4 byte in movimento: questi costituiscono una parte delle locazioni di memoria usate come area di stack dal processore. Nell'area di stack il sistema operativo immagazzina temporaneamente delle informazioni quali, ad esempio, indirizzi di ritorno da subroutine. L'attività di questi byte è dovuta al fatto che un programma sta girando, pre-



riando ora F1 fino ad arrivare all'indirizzo 1020 o poco più (lo si trova con F5) ci si trova in una zona piena di attività. Questa è l'area in cui si adatta l'espansione da 3K. Non essendo inserita l'espansione questa area non è né di RAM né di ROM e l'attività è quindi causata da valori spuri.

Andiamo ora al 4092 tenendo F1 premuto per un poco. (si rammenta che se si dovesse superare il punto preciso si può retrocedere con F3), per chi non volesse aspettare c'è appositamente il tasto —. Nella parte bassa dello schermo si trovano diversi numeri di una, due, o tre cifre. Siamo nell'area di memoria utilizzata dai programmi BASIC ed i numeri che appaiono fanno parte del programma BASIC ora inserito. Se si procede poi fino alla zona occupata dalla memoria di schermo (7680-8191) si avrà un interessante effetto. Lo schermo vede se stesso. È un po' quell'effetto che si ottiene ponendo di fronte due specchi: questi si riflettono l'un l'altro all'infinito. Un'immagine tra i due specchi sarà anch'essa riflessa all'infinito.

Procedendo ora nella scansione fino all'indirizzo 32767, lo schermo mostrerà nella metà alta un tipo di carattere ed in quella bassa un altro carattere.

Questa area (8192-32767) è quella usata dall'espansione RAM: 3 blocchi da 8K ciascuno. Sono così visibili i numeri di pagine di memoria che si stanno attraversando. Ricordiamo che una pagina è formata da 256 byte. Ad esempio, dalla locazione 8192 alla 8447 c'è la pag. 32 (ogni byte di questa conterrà il valore 32 e sarà visibile come bianco). Dalla locazione 8448 alla 8703 c'è la pag. 33 (ogni byte contiene 33 ed è visualizzato come punto esclamativo) e così via.

La parte alta

Andando all'indirizzo 32768 ci si trova all'inizio della ROM caratteri dove il VIC prende le informazioni sui caratteri. Andando dopo la locazione 33800 si ha un cambiamento, poiché sullo schermo appaiono caratteri in reverse: siamo nella zona ROM (33792-33815) contenente le informazioni sui caratteri reversati.

Andiamo nella zona 36864: ci sono qui molti byte che cambiano in continuazione. Questa è una delle aree più interessanti perché siamo nel BLOCCO 0 di input/output. Procedendo attraverso quest'area (36864-38400) si nota molta attività. Questo blocco I/O subisce continue modificazioni: esso tratta an-

che l'I/O di periferiche quali dischi, stampante, etc. Qui si trovano i timer, i registri direzione dati ed altri registri di controllo.

L'ultima e più interessante area visitata dal nostro giro è quella che va da 49152 a 65535. Quello che si vede in quest'area può essere paragonato al cuore del sistema VIC 20. I primi 8K (49152-57343) costituiscono l'interprete BASIC dove il VIC prende tutte le informazioni su come interpretare comandi BASIC come POKE, PRINT e così via. Il secondo gruppo di 8K (57344-65535) costituisce il cervello vero e proprio del calcolatore. Qui risiede il sistema operativo. Quando si scrive un programma, in BASIC o in linguaggio macchina che sia, il S.O. provvede ad eseguire tutte le istruzioni facendo uso anche dell'interprete. È il S.O. che controlla tutto il funzionamento della macchina, esso espleta quindi moltissimi tipi di operazioni. Senza sistema operativo il nostro VIC non sarebbe neppure capace di risolvere una semplice addizione.

Il giro turistico è così terminato, speriamo che a molti sia servito a capire (sia pur a grandi linee) il funzionamento di questa piccola grande macchina che è il VIC 20.

```

10 POKE52,28:POKE56,28:POKE251,0:CLR          :rom 111
20 B=7168:C=7348:X=7196:Y=256:Z=7197          :rem 3
30 FORA=BT0C:READD:POKEA,D:NEXT                :rem 29
40 SYS7168                                       :rem 57
50 PRINT"{HOME}{ 7 SPAZI}";"{HOME}{RED}";      :rem 28
   PEEK(X)+Y*PEEK(Z);"{BLU}"                   :rem 177
60 IFPEEK(251)=1THENGOSUB100                    :rem 128
70 IFPEEK(197)=64THENGOTO40                      :rem 6
80 GOTO50                                         :rem 112
100 POKE251,0:PRINT"{CLR}{GIU'} INDIRIZZO      :rem 112
   DESIDERATO ":POKE198,0                       :rem 112
110 INPUTN                                       :rem 105
120 NN=INT(N/256):POKEX,N-(NN*256):POKEZ,NN    :rem 27
130 POKE7205,PEEK(X):POKE7206,PEEK(Z)+1        :rem 118
150 RETURN                                       :rem 189
7168 DATA162,0,169,6,157,0,150,232           :rem 204
7176 DATA208,250,238,6,28,173,6,28            :rem 132
7184 DATA201,152,208,236,169,150,141,6        :rem 35
7192 DATA28,162,0,189,0,0,157,0               :rem 232
7200 DATA30,232,208,247,189,0,1,157           :rem 81
7208 DATA0,31,232,208,247,165,197,201         :rem 71
7216 DATA64,240,230,162,0,232,208,253         :rem 245
7224 DATA201,39,208,37,24,173,28,28           :rem 240
7232 DATA105,22,141,28,28,173,29,28           :rem 183
7240 DATA105,0,141,29,28,24,173,37             :rem 247
7248 DATA28,105,22,141,37,28,173,38

```

```

7256 DATA28,105,0,141,38,28,76,25           :rem 145
7264 DATA28,201,47,208,37,56,173,28          :rem 253
7272 DATA28,233,22,141,28,28,173,29          :rem 246
7280 DATA28,233,0,141,29,28,56,173           :rem 194
7288 DATA37,28,233,22,141,37,28,173          :rem 252
7296 DATA38,28,233,0,141,38,28,76            :rem 155
7304 DATA25,28,201,55,208,1,96,201           :rem 183
7312 DATA63,208,19,169,0,141,28,28           :rem 194
7320 DATA141,29,28,141,37,28,169,1           :rem 192
7328 DATA141,38,28,76,25,28,201,8            :rem 150
7336 DATA240,3,76,25,28,169,1,133            :rem 144
7344 DATA251,96,0,0,0                        :rem 51

```



NON UN ABBONAMENTO! UN SUPERABBONAMENTO

Questo è ciò che la J.soft ti offre: un superabbonamento ad una super-rivista!

Una eccezionale offerta di lancio: 13 numeri di SUPERVIC al prezzo di 10 oltre alla possibilità esclusiva di ricevere mensilmente, con la copia della rivista, il supporto magnetico con tutti i programmi pubblicati.

L'offerta è valida fino al 20/10/1984.



LE NOSTRE SUPERPROPOSTE ABBONAMENTI

Abbonamento* a 13 numeri (2 nel 1984 + 11 nel 1985) di SUPERVIC al prezzo speciale di L. 35.000

Abbonamento* a 13 numeri di SUPERVIC + 13 cassette con tutti i programmi pubblicati al prezzo speciale di L. 75.000

*L'abbonamento decorrerà dal prossimo numero

Abbonarsi è semplice! Effettuate il versamento con l'apposito modulo c.c.p. inserito in questo fascicolo, oppure ritagliate il tagliando abbonamenti pubblicato in questa pagina e speditelo allegando un assegno intestato a:

J.soft - Via Rosellini 12 - 20124 Milano.

**Tagliando abbonamento a SUPERVIC da inviare in busta chiusa a:
J.soft - Via Rosellini 12 - 20124 Milano**

- ☐ Desidero sottoscrivere un abbonamento a 13 numeri (2 nel 1984 + 11 nel 1985) di SUPERVIC al prezzo speciale di L. 35.000
- ☐ Desidero sottoscrivere un abbonamento a 13 numeri (2 nel 1984 + 11 nel 1985) di SUPERVIC + 13 cassette con tutti i programmi pubblicati al prezzo speciale di L. 75.000

cognome

nome

via

città

cap

provincia

data

firma



Grafica 'Bitmap' per il C64

di M. Tinglot
trad e adatt. di
U. G. Barzaghi

I grafici ad alta risoluzione si ottengono con il metodo 'bitmap'. Questo articolo didattico spiega cosa succede nella memoria del C64 quando utilizzate la tecnica 'bitmap'. Include anche un programma esemplificativo che illustra il metodo in oggetto.

Il C64 permette di realizzare immagini ad alta risoluzione di 320 per 200 punti (chiamati 'pixel'). Per creare queste immagini, il chip VIC-II del C64 usa una tecnica chiamata 'bitmapping'. In parole povere, ciò significa che ciascun bit di una particolare area di memoria rappresenta un pixel (l'unità elementare che appare sul video) dello schermo ad alta risoluzione. Quindi, accendendo o spegnendo particolari pixel, si può formare un disegno.

Potreste chiedere "Perché usare la grafica bitmap quando sono disponibili gli sprite, che sono anche molto più pratici da usare?" La risposta è semplice: ciascun metodo grafico ha le sue possibilità di impiego. Alcune delle principali ragioni a favore dell'impiego dei grafici bitmap sono rappresentate dalla possibilità di creare rappresentazioni grafiche di formule scientifiche e distribuzioni statistiche, disegni a colori ad alta risoluzione, e per creare sfondi ricchi di particolari da utilizzare con gli sprite, ad esempio per un gioco.

Operazioni Binarie

Prima di poter utilizzare efficacemente il metodo bitmap, è importante avere una conoscenza di base della aritmetica binaria (vedi il paragrafo "Aritmetica binaria e bitmapping" in questo stesso articolo) e gli operatori logici AND e OR. Fondamentalmente, essi vengono utilizzati per inizializzare od annullare uno o più bit a scelta in un byte.

AND ed OR provocano un confronto bit-per-bit di due byte per produrne un terzo.

Nel caso di AND, se entrambi i bit sono a 1, il bit risultante è a 1; e nel caso di OR, se entrambi i bit o uno dei due è a 1, il bit risultante, analogamente, è a 1. Ad esempio:

```

10101011      00110001
AND 11011011  OR 10101010
= 10001011    = 10111011
    
```

I bit che compongono un byte sono generalmente numerati nel modo seguente:

7 6 5 4 3 2 1 0

AND viene usato per porre a zero dei bit a scelta, e OR per porli a 1.

Ad esempio:

Problema: porre a zero il bit 5 del byte 10100101. Per fare ciò, definite un byte con il bit 5 posto a 1 (0010000), quindi ottenetene l'inverso (il cui termine appropriato è "complemento"), cambiando tutti gli 1 in 0 e viceversa. Infine, applicate al byte così calcolato

ed al byte dato l'operatore AND:

```

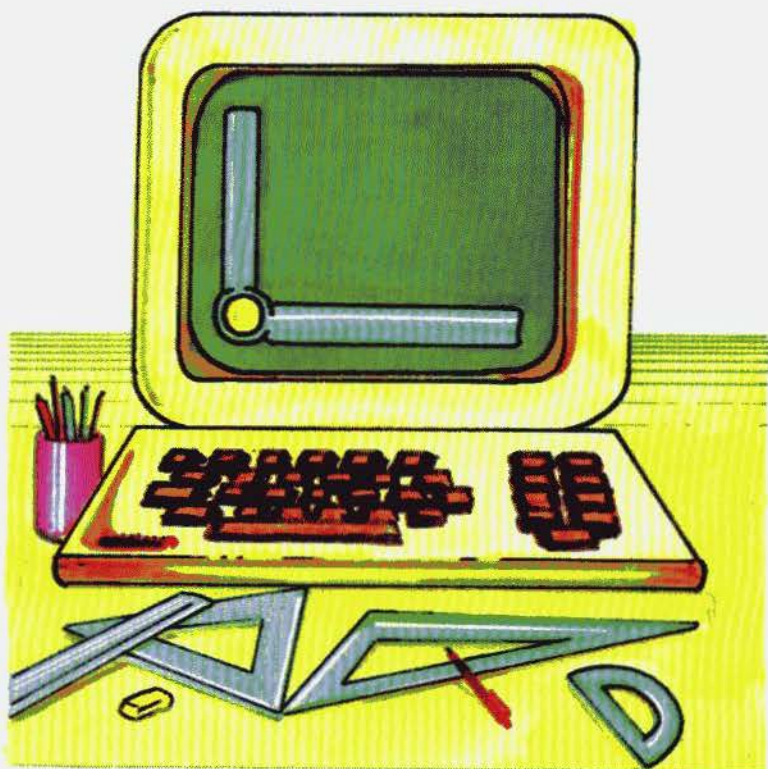
10100101 (dato)
AND 11011111 (calcolato)
= 10000101
    
```

Problema: porre a 1 il bit 6 del byte 10011010. Per fare ciò, definite un byte con il bit 6 posto a 1. Quindi applicate l'operatore OR a questo byte ed a quello dato:

```

10011010 (dato)
OR 01000000 (calcolato)
= 11011010
    
```

Ricordatevi che, quando si utilizza il BASIC, tutti i byte binari devono esse-



re convertiti innanzi tutto in decimale. Le istruzioni OR e AND del BASIC funzioneranno, quindi, come sopra descritto.

Come settare il chip VIC-II

Con una sufficiente comprensione delle istruzioni AND e OR, si può creare un disegno ad alta risoluzione. Il primo passo consiste nel scegliere un'area di memoria di 8000 byte di lunghezza per la bitmap.

Il chip VIC-II accede a soli 16K di memoria contemporaneamente. Dopo l'accensione, il VIC-II vede i primi 16K dalla locazione 0 alla locazione 16383. Tutte le operazioni video, comprese quelle che riguardano la memoria video e la definizione degli sprite, accedono a questa area di memoria. Non c'è spazio in questo blocco di memoria per una bitmap di 8K, in realtà, senza entrare in conflitto con il BASIC. La miglior soluzione è rappresentata dalla scelta di un differente blocco di 16K. (I bit 0 e 1 dell'indirizzo 56576 controllano il posizionamento del blocco all'interno della memoria). Le combinazioni di questi due bit ed il campo di indirizzi che essi rappresentano sono i seguenti:

decimale	binario	indirizzo
0	00	49152-65535
1	01	32768-49151
2	10	16384-32767
3	11	0-16383

Notate come ogni blocco inizi ad un multiplo esatto di 16K. Per scegliere il blocco di memoria per il chip VIC-II, usate il seguente comando:

POKE 56576,Y

dove Y è uno dei valori decimali della tavola precedente. Il blocco più appropriato quando si usa il metodo bitmap unitamente al BASIC è il numero 2: POKE 56576,2

All'interno di questo blocco, devono essere identificate altre due aree: una per gli 8K della bitmap ed una per la memoria video di 1 K. L'indirizzo 53272 viene usato per controllare queste due regioni di memoria. Un bit di questo byte controlla quale sezione di 8K in un blocco di 16K viene usato per la bitmap; quattro bit controllano quale area di 1024 byte viene usata come memoria dello schermo; e tre bit non vengono usati. I bit dell'indirizzo 53272 sono disposti nel modo seguente:

7	6	5	4	3	2	1	0
memoria	bit-				x	x	x
dello	map						
schermo							

Le aree scelte devono essere comprese entro confini finiti - vale a dire, il loro indirizzi di partenza devono essere dei multipli delle loro dimensioni.

Ad esempio, se il blocco di 16K scelto è quello da 0 a 16383, la memoria dello schermo deve iniziare alle locazioni

0 o 1024 o 2048 e così via. La seguente tabella può essere usata per stabilire quale blocco debba essere usato per la memoria dello schermo o per la bitmap:

Indirizzo di base più	blocco mem
0	0
1024	1
2048	2
3072	3
4096	4
5120	5
6144	6
7168	7
8192	8
9216	9
10240	10
11264	11
12288	12
13312	13
14336	14
15360	15

Dove l'indirizzo di base è rappresentato dall'indirizzo della prima locazione del blocco di 16K scelto. Per posizionare opportunamente i bit della locazione 53272, usate la seguente formula:

POKE 53272, blocco di memoria schermo x 16 + blocco bitmap x 8
Se utilizzate la tecnica bitmap ed il BASIC contemporaneamente, usate la seguente istruzione di POKE:

POKE 53272,120
Questa istruzione assegna il settimo blocco di memoria allo schermo, e l'ottavo alla bitmap. Per il blocco di memoria di 16K suggerito per l'uso congiunto con il BASIC, ciò significa che la memoria video parte da 23552 e la bitmap da 24576.

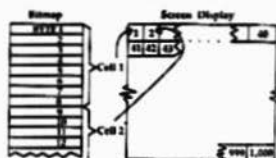
Una volta inizializzati i puntatori alla memoria, il chip VIC-II deve ricevere un comando che lo informi di mostrare la bitmap sul video. Il bit 5 dell'indirizzo 53265 attiva il metodo bitmap, vale a dire, mostra la memoria della bitmap. Per settare questo bit, usate la seguente istruzione POKE:

POKE 53265,PEEK(53265)OR215

Come si disegna

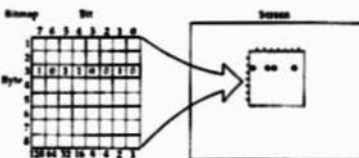
Ora si può creare un disegno in alta risoluzione - tutto ciò che dovete fare consiste nel posizionare a zero o ad uno gli opportuni bit nella memoria bitmap. Il problema consiste nello stabilire quale bit controlla il pixel voluto. Ciò richiede una più profonda comprensione del meccanismo con cui il chip VIC-II traccia la bitmap allo schermo.

La memoria bitmap è organizzata in maniera simile alla memoria di schermo.



mo in modo normale (text mode) - è suddivisa in 1000 aree; ciascuna avente una dimensione di otto byte, che chiameremo 'celle'.

Queste celle sono disposte consecutivamente in memoria - la cella 1 segue la cella 0, la cella 2 segue la cella 1, e così via. Esse sono disposte nella bitmap in un ordine simile a quello dello schermo video in formato normale. 40 celle per riga, 25 righe. L'intero procedimento, così come è stato descritto finora, può essere illustrato come segue:



Ogni cella controlla un'area di 64 pixel organizzata in una matrice 8x8. Il primo byte nella cella controlla la riga superiore di pixel interni a questa matrice, la riga inferiore è controllata dal secondo byte, e così via.

Gli otto bit in ogni byte controllano un singolo pixel in quella riga - il bit più significativo controlla il pixel più a sinistra e così via fino al bit meno significativo, che controlla il pixel alla estremità destra della riga.

Utilizzare le coordinate X ed Y è scomodo con questo sistema. Se è necessario questo tipo di rappresentazione grafica, le equazioni seguenti stabiliscono quale bit attivare per le coordinate X,Y:

Y1 = INT(Y/8) x 8; stabilisce quale riga di celle

X1 = INT(X/8) x 8; stabilisce quale cella della riga suddetta.

AD = Y1 x 320 + X1 + Y.Y1 + indirizzo di partenza della memoria bitmap; stabilisce l'indirizzo del byte opportuno

BT = 7 - X1 stabilisce quale bit attivare POKEAF, PEEK(AD) OR21BT attiva il bit voluto

Se state eseguendo i nostri consigli per quel che riguarda i comandi che sistemano le varie aree di memoria controllate, utilizzate come indirizzo di partenza della bitmap 24576.

Come aggiungere il colore

Il colore rappresenta una parte importante della grafica ad alta risoluzione. Ciascuno dei byte nella memoria di schermo controlla il colore di una cella. Notate che la memoria di schermo controlla il colore 'solo' in bitmap - nel formato normale di testo, contiene il carattere mostrato sullo schermo. I byte nella memoria di schermo sono nello stesso ordine in cui le celle com-



paiono nella bitmap (il colore della cella 650 è controllato dal byte 650 in memoria di schermo). In ogni byte, quattro bit vengono utilizzati per controllare il colore di ogni bit nella corrispondente cella della bitmap, e quattro bit vengono usati per controllare il colore dei bit posti a zero. Questi bit sono organizzati in ogni byte della memoria di schermo come segue:

7	6	5	4	3	2	1	0
colore dei				colore dei			
bit = 1				bit = 0			

colori ed i valori ad essi corrispondenti sono elencati a pag. 139 del "Manuale d'uso". Una volta trovato il valore corrispondente al colore desiderato, usate la seguente formula:
(colore dei bit = 1) x 16 + (colore dei bit = 0)

Inserite questo valore tramite una istruzione POKE nel byte opportuno della memoria di schermo. Ricordatevi che il tentativo di cambiare il colore di un singolo pixel provocherà il cambiamento del colore di tutti i pixel in quella cella della memoria bitmap.

Notate che la memoria dello schermo per il nostro programma esemplificativo inizia all'indirizzo 23552. Ricordatevi che questo metodo può essere utilizzato per creare disegni con 320 x 200 pixel di risoluzione. Tuttavia è disponibile un altro metodo bitmap. Questo secondo metodo consente di avere quattro colori in ogni cella invece che due come detto più sopra. C'è un solo inconveniente: la risoluzione è ridotta a 160 pixel per 200, ed ogni pixel ha dimensioni doppie. Il metodo multicolore viene attivato settando a 1 il bit 4 della locazione 53270. Usate questo comando per attivare il metodo multicolore:

POKE53270,PEEK(53270)OR:4

Ogni pixel è ora rappresentato da due bit. Questi due bit hanno quattro possibili combinazioni, che hanno come risultato quattro colori disponibili. Per trovare il colore che ogni combinazione di bit rappresenta, si accede a parecchie locazioni e/o aree di memoria: la memoria di schermo, la memoria colore (questa è sempre fissata dalle locazioni da 55296 a 55319), ed il registro di colore di fondo alla locazione 53281. La memoria colore è organizzata nello stesso modo della memoria di schermo. La tabella seguente mo-

stra quale combinazione di bit accede a ciascuna area di memoria:

Combinazione di Bit	Colore da
00	registro colore di fondo (53281)
01	memoria di schermo (4 bit più significativi, analogamente ai bit posti ad uno del metodo in due colori)
10	memoria di schermo (4 bit meno significativi, analogamente ai bit posti a zero)
11	memoria colore

Ricordate che tre dei quattro colori scelti possono essere differenti per ogni cella di memoria bitmap. Il metodo seguito per tracciare la bitmap sullo schermo in due colori è utilizzata anche nel metodo multicolore — solo che, ora, i bit sono raggruppati a coppie. Le coppie sono organizzate sequenzialmente, così che il bit 7 ed il bit 6 sono accoppiati, il bit 5 è unito al 4 e così via.

Come proteggere il vostro disegno

Quando usate contemporaneamente il BASIC ed il metodo bitmap, il BASIC potrebbe avere la tendenza di utilizzare la memoria bitmap per ospitare il programma e/o le variabili. Per impedire ciò, cambiate il contenuto degli indirizzi 55 e 56, i byte che segnalano la fine dell'area di memoria riservata al BASIC. Cambiate semplicemente queste due locazioni in modo che puntino ad un indirizzo inferiore all'indirizzo più basso utilizzato. L'indirizzo 56 è uguale alla parte intera della divisione che ha come dividendo l'ultimo indirizzo utile e come divisore 256, mentre l'indirizzo 55 rappresenta il resto della divisione suddetta.

Dopo aver cambiato questi due byte, eseguite una istruzione CLR. Ad esempio questa istruzione assicura che il BASIC non usi celle di memoria al di là dell'indirizzo 23552:
POKE55,0:POKE56,92:CLR
Per ripristinare le condizioni iniziali del vostro 64, usate le istruzioni seguenti:
POKE53265,27:POKE53270,200:
POKE53272,20:POKE56576,151

Aritmetica binaria e bitmapping

Lo schermo del Commodore 64 ad alta risoluzione grafica consiste in 64.000 (320x200) punti o 'pixel'. Ciascuno di essi può essere 'acceso' o 'spento' per

consentirvi di creare grafici personalizzati. Questa tecnica prende il nome di 'bitmapping'.

A prima vista, potreste pensare che, se ci sono 64.000 pixel da controllare, siano necessarie 64.000 locazioni di memoria (byte) — ma ciò impiegherebbe più spazio in memoria di quello effettivamente disponibile. Con la tecnica 'bitmapping', un byte controlla non uno, ma otto pixel! Poiché un byte è composto da otto bit (il bit è l'unità elementare di immagazzinamento di un calcolatore), ogni bit rappresenta un pixel dello schermo ad alta risoluzione.

ne. Quindi, solo 8.000 byte (suppergiù 8K) sono necessari per il bitmapping. Vediamo come il computer gestisce questi bit e questi byte.

Filamenti e lampadine

I calcolatori utilizzano il sistema numerico binario, invece del decimale a cui noi siamo abituati. Un buon metodo per capire il funzionamento del sistema binario consiste nel pensare ad una fila di lampadine, ciascuna delle quali può essere accesa o spenta. La fila ha otto lampadine e rappresenta un byte, ogni lampadina rappresenta un bit. Se sono spente:

00000000

abbiamo un valore numerico zero. Ora accendiamo la lampadina più a destra:

00000001

Ciò dà un valore 1. Fino a questo punto non è affatto difficile.

La lampadina successiva, a partire dalla destra, tuttavia, ha una caratteristica costruttiva completamente differente: ha due filamenti. Se solo questa lampadina è accesa, viene rappresentata come segue:

00000010

ma, ricordatevi, questa lampadina ha due filamenti, quindi il suo valore è 2. Facciamo un passo ed accendiamo anche la prima lampadina:

00000011

Ora abbiamo un valore numerico 3. Due sole lampadine sono accese, ma tre filamenti sono illuminati. La lampadina successiva, la terza dalla destra, contiene quattro filamenti (il doppio della lampadina precedente). Quindi, se viene accesa:

00000100

abbiamo un valore numerico di 4. Se

accendiamo anche le lampadine precedenti:

00000111

abbiamo 7 (4 + 2 + 1) filamenti illuminati, ma solo tre lampadine accese.

Il valore binario 00000111, quindi, equivale al valore decimale 7.

Possiamo cominciare ad intuire qual è il meccanismo che sta alla base della numerazione binaria: ogni lampadina ha il doppio dei filamenti della lampadina alla sua immediata destra:

00000001 = 1	00010000 = 16
00000010 = 2	00100000 = 32
00000100 = 4	01000000 = 64
00001000 = 8	10000000 = 128

Come si realizza una conversione da decimale a binario

Opportune combinazioni di lampadine accese o spente permetterà di ottenere qualsiasi numero compreso tra 0 e 255 (11111111). Scegliamo un numero, diciamo 209, e vediamo come lo si può rappresentare in filamenti, quali lampadine dovremo accendere?

Poiché, accendendo una sola lampadina, ne possiamo illuminare, al massimo, 128, vediamo come farlo:

10000000 (128)

Se aggiungiamo la lampadina alla immediata destra, con 64 filamenti, otterremo un valore numerico di 192 (128 + 64):

11000000 (192)

A questo punto, non possiamo utilizzare la lampadina successiva (con 32 filamenti) perché eccederebbe il numero che vogliamo rappresentare (209): proviamo quindi la successiva, con il 16. Possiamo accenderla poiché ci porterà ancora più vicino al totale che vogliamo rappresentare, ma senza superarlo (192 + 16 = 208):

11010000 (208)

abbiamo bisogno di solo un altro filamento per arrivare a 209, a questo punto è facile, poiché c'è una sola lampadina con un unico filamento, la prima che abbiamo preso in considerazione. Accendiamo questa:

11010001 (209)

ed ora abbiamo 209 filamenti illuminati con solo 4 lampadine accese. Come si può applicare tutto ciò al bitmapping? Il chip VIC-II, scandisce un'area di memoria riservata in esclusiva alla tecnica di bitmapping. Il chip legge ogni bit di ogni byte in quest'area, controllando se sia ad 1 (acceso) o a 0 (spento). Quando si rileva un 1, il pixel che quel bit rappresenta viene illuminato, se si rileva uno 0, il pixel assume lo stesso colore dello sfondo.

Tenendo a mente queste brevi note circa l'aritmetica binaria, date un'occhiata all'articolo in questo stesso numero per vedere come controllare in pratica bit e byte per la tecnica di bitmapping. Vengono anche illustrati comandi speciali per il trattamento delle cifre binarie di cui abbiamo parlato.

Una dimostrazione grafica

Se tutti i calcoli necessari per trovare il bit adatto vi sembrano complicati, e vi accontentate della grafica bicolore, usati il seguente programma di utilità pratica. È stato scritto in linguaggio macchina per accrescerne la velocità, e può essere usato con un sistema di coordinate cartesiane (X,Y). Viene richiamato dal BASIC tramite una istruzione SYS.

Il formato della istruzione SYS è il seguente:

SYS (indirizzo di base del codice in linguaggio macchina), comando, operando/i

I comandi del programma sono i seguenti:

-0=annulla la pagina grafica bitmap (annulla tutti i byte)

-1=assegna il colore allo sfondo. Inizializza tutti i byte dello schermo al va-

lore dell'operando. Ad esempio, SYS(BS),1,32 inizializza ogni byte della memoria di schermo a 32.

-2=traccia un punto. Traccia un punto in base alle sue coordinate cartesiane X,Y. Notate che l'angolo superiore sinistro ha coordinate (0,0) e l'inferiore destro (319,199). Ad esempio, SYS(BS),2,28,122 inserisce un punto in (28,122).

-3=cancella un punto. Il formato del comando è lo stesso del comando precedente.

Questa routine il linguaggio macchina è rilocabile e può essere caricata in qualsiasi parte della memoria semplicemente cambiando il puntatore del programma caricatore BASIC. Prima di poterla utilizzare, assegnate all'indirizzo 680 un valore pari alla parte intera dell'indirizzo di partenza della bitmap diviso per 256. Se avete organizzato la bitmap nel modo suggerito nel nostro esempio funzionante, usate le seguenti istruzioni di POKE:

POKE680,96:POKE81,92

Per vedere come funziona la routine, e qualche altro aspetto della grafica bitmap, provate il programma sottoelencato, che disegna una curva sinusoidale sullo schermo.

La grafica bitmap è una parte consistente delle potenzialità del Commodore 64. Una volta che siate riusciti a padroneggiarla, i risultati possono essere spettacolari. Ricordate, il miglior modo di imparare, consiste nel farlo tramite la pratica diretta. Una volta che vi troviate a vostro agio con la tecnica che abbiamo illustrato, provate qualcuna di queste idee:

1. Disegnate una figura in memoria, quindi spostate i puntatori a quell'area. Ciò permette di far apparire la grafica ad estrema velocità, anche all'interno di programmi BASIC.

2. Utilizzate più di una bitmap e passate i puntatori da una all'altra. Ancora una volta otterrete effetti grafici estremamente veloci.

3. Usate gli sprite. Dal momento che gli sprite sono completamente indipendenti dallo sfondo, potete creare fantastici effetti grafici per i giochi.

```
3 REM COMANDI: :rem 76
4 REM XX SYS (BASE),OPZIONE,DATI:rem 209
5 REM OPZIONI: :rem 123
6 REM SYS B, 0 - CANCELLA LO SCHERMO :rem 195
7 REM SYS B, 1, CL - ASSEGNA IL COLORE DE FINITO DAL VALORE NUMERICO CL :rem 217
8 REM SYS B, 2, X, Y - TRACCIA IL PUNTO (X,Y) :rem 65
9 REM SYS B, 3, X, Y - CANCELLA IL PUNTO (X,Y) :rem 127
10 AD=32768:REM ** INDIRIZZO DI BASE ** :rem 93
20 READ:IFD=-1THEN500:REM ** RICHIAMA LA ROUTINE D'UTENTE :rem 209
30 POKEAD,D:AD=AD-1:GOTO20 :rem 57
```

```
100 DATA32,115,0,32,158,173,32,247,183,14 :rem 122
0,170,2,192,0 :rem 122
110 DATA240,6,192,1,240,32,208,77,173,168 :rem 134
,2,133,252,24 :rem 134
120 DATA105,32,133,253,169,0,133,251,168, :rem 172
145,251,230,251,208 :rem 172
130 DATA2,230,252,166,252,228,253,144,242 :rem 231
,96,32,115,0,32 :rem 231
140 DATA158,173,32,247,183,132,253,173,16 :rem 42
9,2,56,233,1,133 :rem 42
150 DATA252,24,105,4,133,254,169,8,133,25 :rem 88
1,160,247,165,253 :rem 88
160 DATA145,251,230,251,208,2,230,252,166 :rem 228
,252,228,254,144,242 :rem 228
170 DATA96,32,115,0,32,158,173,32,247,183 :rem 187
,140,171,2,141 :rem 187
```



Una grande impresa editoriale

Il micro-millennio è cominciato.

Siamo nell'era dell'elettronica e dell'informatica.

Una rivoluzione silenziosa sta cambiando il nostro modo di vivere, pensare, esprimerci.

Una scelta ci sta oggi davanti: subire le novità che ci attendono oppure viverle da protagonisti: impadronirci del futuro o farcene travolgere. Decidiamo! Varcare le soglie del micro-millennio conoscendone tutti i segreti è oggi possibile.

Oggi c'è E.I. l'enciclopedia dell'elettronica e dell'informatica.

Un'opera unica al mondo, scritta da specialisti per uomini-protagonisti.

E' completa, rigorosa, documentata, facile da capire... anche se parla di elettrotecnica, elettronica allo stato solido, elettronica digitale, microprocessori, comunicazioni, informatica di base, informatica e società.

Tutto quello che volete e dovete sapere sul micro-millennio che ci sta aspettando.



Enciclopedia di Elettronica e Informatica

50 fascicoli settimanali

- 12 pagine di elettronica digitale e microprocessori
- 16 pagine di informatica (oppure elettronica di base e comunicazioni)
- 1 scheda (2 pagine) di elettrotecnica per ottenere in meno di un anno

- 7 grandi volumi
- 1400 pagine complessive
- 1 volume schede di elettrotecnica

L'opera è arricchita da circa 700 foto e 2200 illustrazioni a colori.

Ogni settimana l'elettronica, l'informatica, l'elettrotecnica in un unico fascicolo

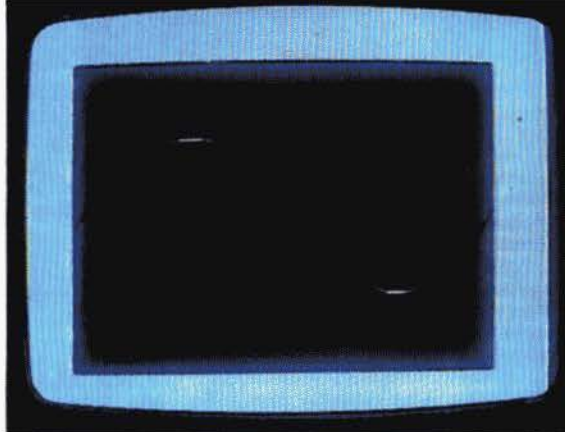


Enciclopedia di Elettronica e Informatica
Oggi in edicola... domani nella vostra biblioteca



GRUPPO EDITORIALE
JACKSON

In collaborazione con il Learning Center
TEXAS INSTRUMENTS



```

180 DATA172,2,32,115,0,32,158,173,32,247,
    183,140,173,2 :rem 133
190 DATA152,41,248,133,253,141,180,2,141,
    174,2,169,0,133 :rem 232
200 DATA254,141,181,2,162,4,24,38,253,38,
    254,202,16,248 :rem 188
210 DATA162,2,24,46,180,2,46,181,2,202,16,
    246,24,165 :rem 30
220 DATA253,109,180,2,141,178,2,165,254,1
    09,181,2,141,179 :rem 37
230 DATA2,173,171,2,41,248,141,176,2,173,
    172,2,141,177 :rem 136
240 DATA2,56,173,173,2,237,174,2,24,109,1
    76,2,133,251 :rem 90
250 DATA173,177,2,109,168,2,133,252,24,17
    3,178,2,101,251 :rem 241
260 DATA133,251,173,179,2,101,252,133,252
    56,173,171,2,237 :rem 83
270 DATA176,2,133,253,56,162,255,169,0,10
    6,232,228,253,208 :rem 94
280 DATA250,141,180,2,174,170,2,224,3,240
    10,160,0,177 :rem 118
290 DATA251,13,180,2,145,251,96,56,169,25
    5,237,180,2,141 :rem 253
300 DATA180,2,160,0,177,251,45,180,2,145,
    251,96,-1 :rem 185
500 REM ** ROUTINE D'UTENTE ** :rem 135
501 REM GRAFICO DI UNA CURVA SINUSOIDALE
    :rem 167
505 POKE53265,PEEK(53265)OR215:REM ** ATT
    IVA IL METODO BIT MAP :rem 48
510 POKE680,96:POKE681,92:REM ** INIZIALI
    ZZA I PUNTATORI ALLA ROUTINE:rem 172
515 POKE53272,120:POKE56576,2:REM ** ATT
    VA LA MEMORIA DEL CHIP VIC-II
    :rem 204
520 POKE55,0:POKE56,60:CLR:REM ** PROTEGG
    E LA BIT MAP DAL PROGRAMMA BASIC
    :rem 255
530 B=32768:REM ** INIZIALIZZA L'INDIRIZZ
    O DI BASE DELLA ROUTINE :rem 70
540 SYS B,0:SYS B,1,16:REM ** CANCELLA LO
    SCHERMO E ASSEGNA IL COLORE:rem 146
550 FORX=0TO6.4STEP.02:Y=SIN(X):REM ** CA
    LCOLA IL VALORE DELLA CURVA SINUSOIDA
    LE :rem 146
560 X1=X*50:Y=Y*50:REM ** AMPLIFICA LE DI
    MENSIONI DEL GRAFICO :rem 72
570 Y=100-Y:SYS B,2,X1,Y:REM ** TRACCIA
    UN PUNTO :rem 254
580 NEXT:REM ** PUNTO SUCCESSIVO:rem 235
590 GOTO590 :rem 117
600 REM ** SI ESCE PREMENDO RUN-STOP/REST
    ORE :rem 159

```

Terapia

di S. Rubio
trad. e adatt. di
di P. Ferrari

Terapia non potrà mai sostituire Freud, ma può aiutarvi a vincere la vostra depressione. Per VIC 20 con espansione e Commodore 64.

"Eliza", il computer psicanalista, è forse il programma più famoso fra quelli che simulano un'intelligenza artificiale. Scritto in LISP da Joseph Weizenbaum nel 1966, negli anni seguenti Eliza è stato utilizzato su computer di tutte le dimensioni e di tutti i tipi, compresi gli home computers programmati in BASIC.

Vi è qualcosa di affascinante nel condurre una conversazione apparentemente razionale con una macchina. Mi ricordo del brivido provato quando scoprii che il mio VIC era in grado di pormi una domanda ("Come ti chiami?"), e ricordo ancora la risposta. È questo stesso motivo che mi ha spinto a scrivere "Terapia".

Un Analista più intelligente

Perché un'altra versione di Eliza? Principalmente per il fatto che se scritto in BASIC, Eliza è estremamente lento e può richiedere anche dieci secondi per rispondere ai vostri commenti. Ho pensato che, come analista, Eliza fosse un po' superato ed anche piuttosto ottuso. Il problema in BASIC è che Eliza esegue un'analisi troppo accurata. L'eseguire una ricerca fra cinquanta parole chiave e cento risposte possibili rende il programma lento; ed il voler fornire dei commenti appropriati a "tutte" le affermazioni dell'utente richiede molto tempo col risultato di ottenere solo dei successi occasionali, anche se spettacolari. Le affermazioni di Eliza possono spesso provocare delle risate od essere causa di irritazione.



Ciò è tuttavia accettabile per il fatto che Weizenbaum non aveva l'intenzione di scrivere un programma che servisse effettivamente come sostituto dell'analista. Ma se volete far vedere il vostro computer agli amici potrebbe

essere divertente avere a disposizione un programma che dimostri l'"intelligenza" della vostra macchina. La versione per il VIC 20 richiede un'espansione di memoria. Qualunque espansione (3K, 8K, ecc...) va bene.

Terapia versione per VIC 20

```
100 PRINTCHR$(142);CHR$(8);CHR$(30):POKE3
    6879,14:GOSUB1230:POKE198,0:rem 174
105 Q=0:QD=0:rem 144
110 PRINTCHR$(147);"SALVE. SONO IL DOTT.
    { 2 SPAZI}ROM.":PRINT"QUAL'E' IL TUO
    NOME?":rem 90
115 GOSUB1160:AS=PI$;PRINT:rem 39
```

```
120 PRINT"IN UNA PAROLA,{ 8 SPAZI}";AS;"
    ":PRINT"QUAL'E' IL TUO":PRINT"PROBLEM
    A?":rem 15
125 GOSUB1160:BS=PI$:rem 98
130 PRINT:PRINTBS;"...?":PRINT:PRINT"PUOI
    DIRMI DI PIU'?" :rem 251
140 GOSUB1160:GOSUB900:rem 48
150 PRINT:PRINT"CAPISCO":PRINTBS:PRINT"E'
    UN GROSSO PROBLEMA PER TE.":rem 84
160 GOSUB1160:IFPI$="NO"THENPRINT"FORSE N
    ON CAPISCO{ 5 SPAZI}BENE...":rem 123
```



Example



```

170 PRINT:PRINT"PUOI ESSERE PIU'
{ 6 SPAZI}PRECISO? PERCHE'":PRINTBS:P
RINT"E' UN PROBLEMA?" :rem 83
180 GOSUB1160:GOSUB900 :rem 52
190 PRINT:PRINT"COME TI FA SENTIRE
{ 4 SPAZI}QUESTO, ":PRINTAS;"?":GOSUB
1160:CS=PLS :rem 144
200 PRINTCHRS(147); :rem 72
205 PRINT"CIO' CHE MI DICI,{ 5 SPAZI}";AS
",":PRINT"E' CHE IL TUO":PRINT"PROBLE
MA CON" :rem 54
210 PRINTBS:PRINT"TI FA SENTIRE":PRINTCS;
".":GOSUB1160 :rem 44
220 PRINT:PRINT"PUOI DIRMI DI PIU'
{ 4 SPAZI}SUI TUOI SENTIMENTI?":GOSUB
1160:GOSUB900 :rem 198
230 PRINT:PRINT"HAI AVUTO LO STESSO
{ 3 SPAZI}PROBLEMA IN PASSATO?
{ 2 SPAZI}(SI O NO)":GOSUB1160
:rem 168
240 IFPLS<>"NO"THEN260 :rem 236
250 PRINT"VEDO. QUESTA NUOVA{ 4 SPAZI}SIT
UAZIONE DEV'ESSERE DIFFICILE PER TE."
:GOTO320 :rem 45
260 PRINT:PRINT"TI SENTIVI ":PRINTCS:PRIN
T"ANCHE ALLORA?" :rem 129
270 GOSUB1160:PRINT"DIMMI DI PIU'."
:rem 44
280 GOSUB1160:GOSUB900 :rem 53
290 PRINTCHRS(147)"PENSO CHE STIAMO
{ 6 SPAZI}ARRIVANDO A QUALCOSA. CREDI
CHE CI SIA UN" :rem 45
295 PRINT"LEGAME?" :rem 91
300 GOSUB1160:PRINT:PRINT"CONTINUA..."
:rem 152
310 GOSUB1160:PRINT:PRINT"SEMBRA CHE SIA
UN{ 5 SPAZI}GROSSO PROBLEMA PER
{ 3 SPAZI}TE.":GOSUB1160 :rem 23
320 PRINT:PRINT"HAI QUALCHE IDEA PER
{ 2 SPAZI}RISOLVERE QUESTA{ 6 SPAZI}
SITUAZIONE?" :rem 212

330 PRINT"(SI O NO)":GOSUB1160 :rem 194
340 IFPLS<>"SI"THEN350 :rem 236
343 PRINT"PENSI CHE QUESTA IDEA AVRA' SUC
CESSO?:GOTO360 :rem 164
350 PRINT:PRINT"PERCHE' NON FAI UN
{ 4 SPAZI}ELENCO DELLE POSSIBILISOLUZ
IONI?" :rem 65
360 GOSUB1160:GOSUB900 :rem 52
370 FORI=1TO500:NEXTI:PRINTCHRS(147);
:rem 56
380 PRINT"OKAY, QUAL'E' LA{ 6 SPAZI}PAROL
A CHE DESCRIVE{ 3 SPAZI}MEGLIO I TUOI
" :rem 74
385 PRINT"SENTIMENTI IN QUESTO{ 2 SPAZI}
MOMENTO?" :rem 71
390 GOSUB1160:DS=PLS:PRINT:PRINTDS;"...?"
:rem 166
400 GOSUB1160:GOSUB900:PRINT :rem 246
410 PRINT"ORA POTREMMO FARE{ 5 SPAZI}QUAL
COS'ALTRO.":PRINT"PROVIAMO CON"
:rem 32
430 PRINT"L'ASSOCIAZIONE DI{ 5 SPAZI}PARO
LE E VEDIAMO DOVE CI PORTA.":rem 107
440 PRINT"COSA NE PENSI":PRINT"(SI O NO)?
":GOSUB1160 :rem 70
450 IFPLS="SI"THEN490 :rem 182
460 PRINT:PRINT"MI SEMBRA CHE CIO' TI CRE
I DEI PROBLEMI." :rem 110
470 PRINT"PUOI PARLARMENE?":GOSUB1160:IFP
LS="NO"THEN830 :rem 167
480 PRINT:PRINT"PENSO CHE":PRINT"L'ASSOCI
AZIONE DI{ 5 SPAZI}PAROLE SIA MOLTO
{ 6 SPAZI}UTILE." :rem 248
490 PRINT:PRINT"PROVIAMO." :rem 208
500 PRINT"IO DIRO' UNA PAROLA{ 3 SPAZI}E
TU DIRAI LA PRIMA{ 3 SPAZI}PAROLA CHE
TI VIENE{ 3 SPAZI}IN MENTE":rem 6
510 REM ***ASSOCIAZIONE DI PAROLE***
:rem 78
520 FORI=1TO5000:NEXTI:PRINTCHRS(147);"CA
NE":PRINT:GOSUB1160 :rem 9
530 PRINT:PRINT"BERE":PRINT:GOSUB1160
:rem 151
540 PRINT:PRINT"CASA":PRINT:GOSUB1160:ES=
PLS :rem 23
550 PRINT:PRINTBS:PRINT:GOSUB1160:FS=PLS
:rem 35
560 PRINT:PRINT"SENTIMENTI":PRINT:GOSUB11
60 :rem 124
570 PRINT:PRINT"DIVERTIRSI":PRINT:GOSUB11
60:GS=PLS :rem 9
580 PRINT:PRINT"MAMMA":PRINT:GOSUB1160:IS
=PLS :rem 112
590 PRINT:PRINTCS:PRINT:GOSUB1160:JS=PLS
:rem 44
600 FORI=1TO1000:NEXTI:PRINTCHRS(147)
:rem 37
610 PRINT"HO NOTATO CHE QUANDO{ 2 SPAZI}
HO DETTO CASA TU HAI{ 2 SPAZI}DETTO "
;ES;".":rem 231
620 PRINT"CIO' RIFLETTE IN{ 6 SPAZI}QUALC
HE MODO I TUOI{ 3 SPAZI}SENTIMENTI VE
RSO TE{ 3 SPAZI}STESSO?" :rem 101
630 PRINT"SI O NO":GOSUB1160:IFPLS<>"SI"
HEN650 :rem 6
640 PRINT:PRINT"IN CHE MODO?":GOSUB1160:G
OSUB900 :rem 220
650 PRINT:PRINT"CHE RELAZIONE C'E'
{ 4 SPAZI}COL TUO PROBLEMA CON ":PRIN
TBS :rem 183

```

```

660 GOSUB1160:GOSUB900:PRINT:PRINT"QUANDO
    HO DETTO":PRINTBS:PRINT"TU HAI DETTO
    ":PRINTFS      :rem 76
670 PRINT"COSA PENSI CHE":PRINT"SIGNIFICHI
    I QUESTO?":GOSUB1160:GOSUB900      :rem 192
680 PRINT:PRINT"SEI STANCO? VUOI UN
    { 3 SPAZI}KLEENEX?":GOSUB1160      :rem 209
690 IF PLS<>"SI"THEN710      :rem 244
700 PRINT"ECCO.":FORT=1TO1000:NEXTT      :rem 196
710 PRINT:PRINT"E' INTERESSANTE CHE
    { 3 SPAZI}QUANDO HO DETTO      :rem 190
715 PRINT"DIVERTIRSI, TU HAI{ 4 SPAZI}DET
    TO ":PRINTGS      :rem 204
720 GOSUB1160:GOSUB900:PRINTCHRS(147);"HM
    MMM..."      :rem 110
730 PRINT:PRINT"MI SEMBRA CHE TUTTO
    { 3 SPAZI}CIO', ":PRINTAS;"      :rem 247
735 PRINT"SIA LEGATO AL TUO{ 5 SPAZI}PROB
    LEMA CON":PRINTBS;"      :rem 154
750 GOSUB1160:GOTO770      :rem 245
760 REM ***SOGLI***      :rem 253
770 PRINT:PRINT"PROVIAMO CON UN
    { 7 SPAZI}APPROCCIO DIVERSO,":PRINTAS
    ;"      :rem 231
780 PRINT"RACCONTAMI UNO DEI{ 4 SPAZI}TUO
    I SOGLI.":GOSUB1160:GOSUB1040:IFQD=1T
    HEN840      :rem 156
790 PRINT:PRINT"COME DESCRIVERESTI I
    { 2 SPAZI}TUOI SENTIMENTI NEL
    { 3 SPAZI}SOGLIO?"      :rem 102
791 GOSUB1160      :rem 233
800 PRINT:PRINT"IL SOGLIO AVEVA QUALCHEREL
    AZIONE CON":PRINTIS      :rem 126
810 GOSUB1160:FORT=1TO1000:NEXTT:rem 245
820 REM ***FINITO***      :rem 67
830 PRINTCHRS(147);"PENSO CHE STIAMO
    { 6 SPAZI}PROCEDENDO NELLA{ 6 SPAZI}
    GIUSTA DIREZIONE."      :rem 170
835 PRINT      :rem 45
840 PRINT"ABBIAMO DISCUSO DEL{ 2 SPAZI}
    TUO PROBLEMA CON":PRINTBS:PRINT"E DI
    COME CIO' TI FA"      :rem 24
850 PRINT"SENTIRE ":PRINTCS;"      :rem 112
860 PRINT"E PARLATO DELLE{ 7 SPAZI}POSSIB
    ILI SOLUZIONI."      :rem 213
870 PRINT:PRINT"MA VEDO CHE IL TUO
    { 4 SPAZI}TEMPO E' SCADUTO.":rem 130
875 PRINT"ARRIVEDERCI ALLA{ 6 SPAZI}PROSS
    IMA SETTIMANA."      :rem 1
880 END      :rem 119
890 REM ***PAROLE CHIAVE***      :rem 244
900 IF Q>0THENRETURN      :rem 246
910 FORJ=1TOLEN(PLS)-10      :rem 63
920 IFMIDS(PLS,J,10)<>"DIVERTENTE"THEN930
      :rem 164
925 PRINT:PRINT"COSA PENSI DEL{ 8 SPAZI}
    DIVERTIMENTO?":GOTO 950      :rem 156
930 NEXTJ      :rem 37
940 RETURN      :rem 125
950 GOSUB1160:Q=1:PRINT:PRINT"QUESTI SENT
    IMENTI MI{ 2 SPAZI}SEMBRANO IMPORTANT
    I."      :rem 174
960 GOSUB1160:RETURN      :rem 1
1040 REM ***RICERCA DELLA PAROLA CHIAVE N
    EL SOGLIO***      :rem 212
1050 FORJ=1TOLEN(PLS)-4      :rem 62
1060 IFMIDS(PLS,J,4)="NON "THEN1120
      :rem 127
1070 NEXTJ      :rem 81
1110 RETURN      :rem 163
1120 PRINTCHRS(147)"PERCHE' PENSI CHE SIA
    COSI'?:GOSUB1160:GOSUB900:rem 246
1130 PRINTCHRS(147);"POTREMMO DISCUTERE D
    I CIO' IN SEGUITO."      :rem 78
1140 PRINT"POTREBBE ESSERCI UNA
    { 2 SPAZI}RELAZIONE CON IL TUO"
      :rem 242
1150 PRINT"PROBLEMA CON ":PRINTBS;"":QD=
    1:RETURN      :rem 251
1160 REM ***ROUTINE DI INPUT***      :rem 235
1170 PLS=""      :rem 239
1180 GETP2$:IFP2$=""THEN1180      :rem 57
1190 PRINTP2$;      :rem 57
1200 IF P2$=CHRS(13)THENRETURN      :rem 250
1210 PLS=PLS+P2$      :rem 28
1220 GOTO1180      :rem 200
1230 REM ***INTRODUZIONE***      :rem 80
1240 PRINTCHRS(147);TAB(7)"TERAPIA"
      :rem 38
1250 PRINT:PRINT"VUOI UN'INTRODUZIONE? (S
    /N)"      :rem 119
1260 GETQS:IFQS<>"S"ANDQS<>"N"THEN1260
      :rem 185
1270 IFQS="N"THENRETURN      :rem 172
1280 PRINTCHRS(147);"BENVENUTO ALLA TUA
    { 4 SPAZI}SEDUTA DI ANALISI."
      :rem 1
1285 PRINT"IL DOTT. ROM SARA' DA TE FRA U
    N MOMENTO."      :rem 194
1290 PRINT"MENTRE ASPETTI, ECCO
    { 2 SPAZI}ALCUNI CONSIGLI UTILI SU C
    OME SPRUTTARE NEL"      :rem 206
1300 PRINT"MODO MIGLIORE LA TUA
    { 2 SPAZI}SEDUTA DI ANALISI."
      :rem 21
1310 PRINT"COME LA MAGGIOR PARTE DELLE CO
    SE NELLA VITA,ANCHE NELL'ANALISI"
      :rem 215
1320 PRINT"PIU' DAI, PIU' { 8 SPAZI}RICEVE
    RAT.":PRINT"PUOI PENSARE CHE SIA"
      :rem 134
1330 PRINT"DIVERTENTE INGANNARE
    { 2 SPAZI}IL DOTTOR, RIDERE
    { 4 SPAZI}DELLA SUA GRAMMATICA"
      :rem 227
1331 PRINT"OD INSULTARLO."      :rem 103
1332 PRINT:PRINTCHRS(18)"PREMI UN TASTO"
      :rem 199
1335 POKE198,0:WAIT198,1      :rem 103
1340 PRINT"[CLR]{GIU'}TUTTAVIA, ANCHE SE
    { 4 SPAZI}QUESTO E' UN GIOCO DA SALO
    TTO, POTRESTI"      :rem 132
1350 PRINT"RICAVARNE DEI":PRINT"SUGGERIME
    NTI":PRINT"INTERESSANTI E PERFINOIMP
    ORTANTI."      :rem 214
1360 PRINT"CIO' PUO' AVVENIRE{ 4 SPAZI}SO
    LO SE FARAI DEL TUO MEGLIO PER UTILI
    ZZARE"      :rem 212
1370 PRINT"QUESTA SEDUTA COME UN MODO DIV
    ERTENTE PER{ 3 SPAZI}PONDERARE SUI P
    ROBLEMI";      :rem 50
1380 PRINT"E SULLE SECCATURE{ 5 SPAZI}DEL
    LA VITA."      :rem 202
1390 PRINT:PRINTCHRS(18)"PREMI UN TASTO"
      :rem 203
1400 POKE198,0:WAIT198,1      :rem 96
1410 PRINTCHRS(147):PRINT"IL DOTTOR E'
    { 9 SPAZI}ARRIVATO."      :rem 4

```

```

1420 PRINT:PRINT"PER PARLARE COL DOTT. RO
M, BASTA SCRIVERE{ 3 SPAZI}LA RISPOS
TA ALLE SUE" :rem 137
1430 PRINT"DOMANDE E PREMERE{ 5 SPAZI}";C
HRS(18);"RETURN";CHRS(146);" QUANDO
HAI" :rem 124
1440 PRINT"FINITO.":PRINT:PRINT"DIVERTITI
CON LA TUA{ 2 SPAZI}SEDUTA DI ANALI
SI." :rem 240
1450 PRINTSPC(110);CHRS(18);"PREMI UN TAS
TO PER{ 4 SPAZI}COMINCIARE":rem 1
1460 POKE198,0:WAIT198,1:RETURN :rem 128

```

Terapia versione per C64

```

ROMA: SONO IL DOTT. RO
MANO. IL TUO NOME?
CHIAMAMI.
IN UNA PAROLA, RISPONDI
DOMANDE E PREMERE
5 SPAZI.
QUANDO HAI
FINITO.
DIVERTITI
CON LA TUA
SEDUTA DI ANALI
SI.
PREMI UN TASTO
PER COMINCIARE.

```

```

100 PRINTCHRS(142)CHRS(8)CHRS(30):POKE532
81,0:POKE53280,0:GOSUB1230:POKE198,0
:rem 188
105 Q=0:QD=0 :rem 144
110 PRINTCHRS(147);"SALVE. SONO IL DOTT.
ROM.":PRINT"QUAL'E' IL TUO NOME?"
:rem 90
115 GOSUB1160:AS=PLS:PRINT :rem 39
120 PRINT"IN UNA PAROLA, ";AS;":PRINT"Q
UAL'E' IL TUO PROBLEMA?":GOSUB1160
:rem 134
125 BS=PLS :rem 224
130 PRINT:PRINTBS;"...?":PRINT:PRINT"PUOI
DIRMI DI PIU'?" :rem 251
140 GOSUB1160:GOSUB900 :rem 48
150 PRINT:PRINT"CAPISCO ";BS:PRINT"E' UN
GROSSO PROBLEMA PER TE." :rem 200
160 GOSUB1160:IFPLS<>"NO"THENPRINT"FORSE N
ON CAPISCO BENE..." :rem 123
170 PRINT:PRINT"PUOI ESSERE PIU' PRECISO?
PERCHE'":PRINTBS :rem 168
175 PRINT"E' UN PROBLEMA PER TE?"
:rem 142

```

```

180 GOSUB1160:GOSUB900 :rem 52
190 PRINT:PRINT"QUESTO COME TI FA SENTIRE
, ";AS;"?":GOSUB1160:CS=PLS :rem 4
200 PRINTCHRS(147) :rem 13
205 PRINT"CIO' CHE STAI DICENDO, ";AS;":
PRINT"E' CHE IL TUO PROBLEMA CON ";B
S :rem 127
210 PRINT"TI FA SENTIRE ";CS;":GOSUB116
0 :rem 115
220 PRINT:PRINT"PUOI SPIEGARMI MEGLIO CIO
' CHE SENTI?":GOSUB1160:GOSUB900
:rem 48
230 PRINT:PRINT"HAI AVUTO LO STESSO PROBL
EMA IN PASSATO?(SI O NO)":GOSUB1160
:rem 168
240 IFPLS<>"NO"THEN260 :rem 236
250 PRINT"VEDO. PERCIO' QUESTA NUOVA SITU
AZIONE{ 3 SPAZI}DEV'ESSRE DIFFICILE P
ER TE." :rem 201
255 GOTO 320 :rem 106
260 PRINT:PRINT"TI SENTIVI ";CS;" ANCHE A
LLORA?" :rem 105
270 GOSUB1160:PRINT"DIMMI QUALCOSA DI PIU
'." :rem 133
280 GOSUB1160:GOSUB900 :rem 53
290 PRINTCHRS(147)"PENSO CHE STIAMO ARRIV
ANDO A QUALCOSA." :rem 234
300 PRINT"CREDI CHE CI SIA UN LEGAME?":GO
SUB1160:PRINT:PRINT"CONTINUA..."
:rem 208
310 GOSUB1160:PRINT:PRINT"QUESTO SEMBRA E
SSERE UN GROSSO PROBLEMA"; :rem 29
315 PRINT" PER TE":GOSUB1160 :rem 108
320 PRINT:PRINT"HAI QUALCHE IDEA PER RISOL
VERE QUESTA{ 3 SPAZI}SITUAZIONE?"
:rem 212
330 PRINT"SI O NO.":GOSUB1160 :rem 159
340 IFPLS<>"SI"THEN350 :rem 236
343 PRINT"PENSI CHE QUESTA IDEA POSSA AVE
RE":PRINT"SUCCESSO?":GOTO360:rem 87
350 PRINT:PRINT"PERCHE' NON FAI UN ELENCO
DELLE" :rem 136
355 PRINT"POSSIBILI SOLUZIONI, ALLORA?"
:rem 14
360 GOSUB1160:GOSUB900 :rem 52
370 PORT=10500:NEXTT:PRINTCHRS(147)
:rem 253
380 PRINT"OKAY, QUAL'E' LA PAROLA CHE MEG
LIO" :rem 107
385 PRINT"DESCRIVE COME TI SENTI IN QUEST
O":PRINT"MOMENTO?" :rem 235
390 GOSUB1160:DS=PLS:PRINT:PRINTDS;"...?"
:rem 166
400 GOSUB1160:GOSUB900:PRINT :rem 246
410 PRINT"VEDIAMO COSA POSSIAMO FARE.":PR
INT"PROVIAMO CON L'ASSOCIAZIONE DI ";
:rem 99
430 PRINT"PAROLE":PRINT"E VEDIAMO DOVE CI
PORTA." :rem 238
440 PRINT"COSA NE PENSI (SI O NO)?":GOSUB
1160 :rem 59
450 IFPLS<"SI"THEN490 :rem 182
460 PRINT:PRINT"MI SEMBRA CHE CIO' TI CRE
I DEI PROBLEMI." :rem 110
470 PRINT"PUOI DIRMI IL MOTIVO?":GOSUB1
160:IFPLS<"NO"THEN830 :rem 55
480 PRINT:PRINT"IO PENSO CHE L'ASSOCIAZIO
NE DELLE PAROLE SIA MOLTO UTILE."
:rem 94
490 PRINT:PRINT"PROVIAMO." :rem 208

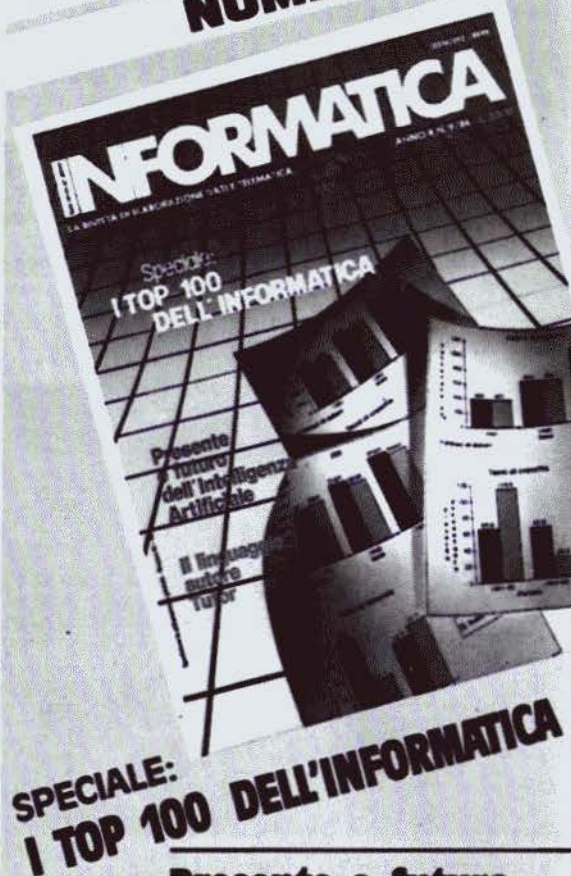
```

```

500 PRINT"IO DIRO' UNA PAROLA. TU DEVI DI
RE LA{ 4 SPAZI}PRIMA PAROLA CHE TI VI
ENE "; rem 253
505 PRINT"IN MENTE." rem 169
510 REM ***ASSOCIAZIONE DI PAROLE*** rem 113
rem 78
520 FOR=1TO5000:NEXTT:PRINTCHR$(147);"CA
NE":PRINT:GOSUB1160 rem 9
530 PRINT:PRINT"BERE":PRINT:GOSUB1160 rem 151
540 PRINT:PRINT"CASA":PRINT:GOSUB1160:ES=
PIS rem 23
550 PRINT:PRINTBS:PRINT:GOSUB1160:FS=PIS rem 35
560 PRINT:PRINT"SENTIMENTI":PRINT:GOSUB11
60 rem 124
570 PRINT:PRINT"DIVERTIRSI":PRINT:GOSUB11
60:GS=PIS rem 9
580 PRINT:PRINT"MAMMA":PRINT:GOSUB1160:IS
=PIS rem 112
590 PRINT:PRINTCS:PRINT:GOSUB1160:JS=PIS rem 44
600 FOR=1TO1000:NEXTT:PRINTCHR$(147) rem 37
610 PRINT"HO NOTATO CHE QUANDO HO DETTO C
ASA":PRINT"TU HAI RISPOSTO ";ES;". rem 245
620 PRINT"FORSE CIO' RIFLETTE IN QUALCHE
MODO I" rem 91
625 PRINT"TUOI SENTIMENTI VERSO TE STESSO
?" rem 247
630 PRINT"(SI O NO):GOSUB1160:IFPIS<>"SI
"THEN650 rem 87
640 PRINT:PRINT"IN CHE MODO?":GOSUB1160:G
OSUB900 rem 220
650 PRINT:PRINT"CHE RELAZIONE C'E' COL TU
O PROBLEMA":PRINT"CON ";BS rem 54
660 GOSUB1160:GOSUB900:PRINT:PRINT"QUANDO
IO HO DETTO ";BS rem 33
665 PRINT"HAI RISPOSTO ";FS rem 108
670 PRINT"CHE SIGNIFICATO DAI A QUESTO?":
GOSUB1160:GOSUB900 rem 114
680 PRINT:PRINT"SEI STANCO? VUOI UN KLEEN
EX?":GOSUB1160 rem 209
690 IF PIS<>"SI"THEN710 rem 244
700 PRINT"ECCO.":FOR=1TO1000:NEXTT rem 196
710 PRINT:PRINT"E' INTERESSANTE CHE QUAND
O HO DETTO" rem 224
715 PRINT"DIVERTIRSI, TU HAI DETTO ";GS rem 64
720 GOSUB1160:GOSUB900:PRINTCHR$(147);"HM
MMM..." rem 110
730 PRINT:PRINT"MI SEMBRA, ";AS;". rem 249
735 PRINT"CHE TUTTO CIO' SIA LEGATO AL PR
OBLEMA" rem 90
740 PRINT"CON ";BS rem 237
750 GOSUB1160:GOTO770 rem 245
760 REM ***SOGLI*** rem 253
770 PRINT:PRINT"PROVIAMO CON UN APPROCCIO
DIVERSO":PRINTAS rem 14
780 PRINT"RACCONTAMI UNO DEI TUOI SOGLI."
:GOSUB1160:GOSUB1040:IFQD=1THEN840 rem 156
790 PRINT:PRINT"COME DESCRIVERESTI I TUOI
SENTIMENTI NELSOGLIO?" rem 102
791 GOSUB1160 rem 233
800 PRINT:PRINT"IL SOGLIO HA AVUTO QUALCOS
A A CHE FARE{ 3 SPAZI}CON ";IS rem 115
810 GOSUB1160:FOR=1TO1000:NEXTT:rem 245
820 REM ***FINITO*** rem 67
830 PRINTCHR$(147);"PENSO CHE STIAMO PROC
EDENDO NELLA GIUSTADIREZIONE.":PRINT rem 113
840 PRINT"ABBIAMO DISCUSO DEL TUO PROBLE
MA CON":PRINTBS rem 241
845 PRINT"E DI COME TI FACCIA SENTIRE" rem 182
850 PRINTCS;";" rem 75
860 PRINT"E DISCUSO ALCUNE SOLUZIONI POS
SIBILI." rem 129
870 PRINT:PRINT"MA VEDO CHE IL TEMPO A TU
A DISPOSIZIONE E' SCADUTO." rem 85
875 PRINT"ARRIVEDERCI ALLA PROSSIMA SETTI
MANA." rem 1
880 END rem 119
890 REM ***PAROLE CHIAVE*** rem 244
900 IF Q>0THENRETURN rem 246
910 FORJ=1TOLEN(PIS)-10 rem 63
920 IFMIDS(PIS,J,10)<>"DIVERTENTE"THEN930 rem 164
925 PRINT:PRINT"COSA PENSI DEL DIVERTIMEN
TO?":GOTO 950 rem 156
930 NEXTJ rem 37
940 RETURN rem 125
950 GOSUB1160:Q=1:PRINT:PRINT"MI SEMBRANO
DEI SENTIMENTI IMPORTANTI." rem 165
960 GOSUB1160:RETURN rem 1
1040 REM ***RICERCA DELLE PAROLE CHIAVE N
EL SOGLIO*** rem 220
1050 FORJ=1TOLEN(PIS)-4 rem 62
1060 IFMIDS(PIS,J,4)="NON "THEN1120 rem 127
1070 NEXTJ rem 81
1110 RETURN rem 163
1120 PRINTCHR$(147)"PERCHE' CREDI CHE SIA
COSI'?:GOSUB1160:GOSUB900:rem 222
1130 PRINT"POTREBBE ESSERE UTILE DISCUTER
NE IN" rem 192
1140 PRINT"SEGUITO. POTREMMO SCOPRIRE CHE
VI E'" rem 154
1150 PRINT"UNA RELAZIONE COL TUO PROBLEMA
CON{ 6 SPAZI}";BS;".QD=1:RETURN rem 151
1160 REM ***ROUTINE DI INPUT*** rem 235
1170 PIS="" rem 239
1180 GETP2S:IFP2S=""THEN1180 rem 57
1190 PRINTP2S; rem 57
1200 IF P2S=CHR$(13)THENRETURN rem 250
1210 PIS=PIS+P2S rem 28
1220 GOTO1180 rem 200
1230 REM ***INTRODUZIONE*** rem 80
1240 PRINTCHR$(147);TAB(15)"TERAPIA" rem 85
1250 PRINT:PRINT"VUOI UN'INTRODUZIONE? (S
/N)" rem 119
1260 GETQS:IFQS<>"S"ANDQS<>"N"THEN1260 rem 185
1270 IFQS="N"THENRETURN rem 172
1280 PRINTCHR$(147);"BENVENUTO ALLA TUA S
EDUTA DI ANALISI.{ 3 SPAZI}IL DOTT.
ROM"; rem 40
1285 PRINT" SARA' DA TE IN UN "; rem 130
1290 PRINT"MENTE. MENTRE ASPETTI, ECCO
ALUNI UTILI" rem 21
1300 PRINT"CONSIGLI SU COME SFRUTTARE NEL
MIGLIOR" rem 107
1305 PRINT"MODI POSSIBILI QUESTA SEDUTA."
:PRINT:PRINT rem 200

```

**E' IN EDICOLA
IL NUOVO
NUMERO DI**



**SPECIALE:
I TOP 100 DELL'INFORMATICA**

**Presente e futuro
dell'Intelligenza
artificiale**

Il Linguaggio Tutor



Una pubblicazione
firmata...

GRUPPO EDITORIALE JACKSON S.p.A.
20124 Milano - Via Rosellini, 12 - Tel. 02/68.80.951-2-3-4-5
Telex: 333436 GEJ IT

LOAD

```

1310 PRINT"COME PER LA MAGGIOR PARTE DELL
    E COSE( 4 SPAZI)NELLA VITA, ANCHE CO
    N "; :rem 134
1320 PRINT"L'ANALISI PIU' ( 4 SPAZI)DAI, P
    IU' RICEVERAI. POTRESTI TROVARE"
        :rem 148
1325 PRINT"DIVERTENTE"; :rem 209
1330 PRINT"INGANNARE IL DOTTORE, RIDERE
    DELLA SUA GRAMMATICA O INSULTARLO."
        :rem 115
1340 PRINT"{GIU'}TUTTAVIA, ANCHE SE SI TR
    ATTA SOLO DI UN GIOCO DA SALOTTO, PO
    TREBBE"; :rem 238
1350 PRINT" DARTI DEI( 4 SPAZI)SUGGERIMEN
    TI INTERESSANTI E PERFINO( 5 SPAZI)
    IMPORTANTI." :rem 153
1360 PRINT"CIO' PUO' ACCADERE SE FARAI DE
    L TUO( 5 SPAZI)MEGLIO PER UTILIZZARE
    "; :rem 154
1370 PRINT" QUESTA SEDUTA COMEUN MODO DIV
    ERTENTE PER PONDERARE SUI" :rem 157
1380 PRINT"PROBLEMI E LE SECCATURE DELLA
    VITA." :rem 48
1390 PRINT:PRINT:PRINTCHR$(18)"PREMI UN T
    ASTO PER CONTINUARE" :rem 113
1400 POKE198,0:WAIT198,1 :rem 96
1410 PRINTCHR$(147):PRINT:PRINT"IL DOTTOR
    E E' ARRIVATO." :rem 203
1420 PRINT:PRINT:PRINT"PER PARLARE COL DO
    TT. ROM BASTA CHE TU" :rem 45
1430 PRINT"SCRIVA LE RISPOSTE E PREMA ";C
    HRS(18);"RETURN";CHRS(146) :rem 162
1435 PRINT"QUANDO HAI FINITO." :rem 47
1440 PRINT:PRINT:PRINT"DIVERTITI CON LA T
    UA SEDUTA DI ANALISI." :rem 181
1450 PRINTSPC(240);CHRS(18);"PREMI UN TAS
    TO PER COMINCIARE" :rem 5
1460 POKE198,0:WAIT198,1:RETURN :rem 128

```

**leggete
PAPER**

OFFERTISSIMA --EXELCO--



**UN RISPARMIO DI OLTRE
250'000 LIRE**

La divis. EXELCO vi propone
una vantaggiosissima
combinazione **COMMODORE**

**Affrettatevi
è un'OFFERTA irripetibile !!!**

**SUBITO
A CASA VOSTRA**

n. 1 COMMODORE C64	L. 699.500
n. 1 Registratore compatibile	L. 82.000
n. 1 Libro "C64 EXPOSED"	L. 24.000
n. 1 Libro "Impariamo il computer"	L. 25.000
n. 1 Cassetta Software "Garden Wars"	L. 20.000

Totale L. ~~850.500~~

A SOLE L. 599'000

IVA INCLUSA

NON PERDETE QUESTA OCCASIONE !

Descrizione	Q.tà	Prezzo unitario	Prezzo Totale
n. 1 COMMODORE C64 n. 1 Registratore compatibile n. 1 Libro "C64 EXPOSED" n. 1 Libro "Impariamo il computer" n. 1 Cassetta Software "Garden Wars"		L. 599.000	

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

SPAZIO RISERVATO ALLE AZIENDE - SI RICHIEDE L'EMISSIONE DI FATTURA

Partita IVA

PAGAMENTO:

A) Anticipato, mediante assegno circolare o vaglia postale per l'importo totale dell'ordinazione.
B) Contro assegno, in questo caso, è indispensabile versare l'acconto di Lire 50.000 mediante assegno circolare o vaglia postale. Il saldo sarà regolato contro assegno.
AGGIUNGERE: L. 5.000 per contributo fisso. I prezzi sono comprensivi di I.V.A.

DIVIS. EXELCO

Via G. Verdi, 23/25
20095 - CUSANO MILANINO - Milano

SI ACCETTANO FOTOCOPIE DI QUESTO MODULO D'ORDINE



di V. Aglio

Quando bisogna fare una scritta gigante per abbellire il nostro programma, il più delle volte si è costretti a scrivere una quantità enorme di PRINT e di segni grafici per poter ottenere una lettera o un numero tre o quattro volte più grande del normale, ed in molti casi con un risultato non troppo soddisfacente; quindi ho pensato che una routine in l/m avrebbe potuto risolvere il problema, e per evitare che una lettera o un numero fosse ottenuta con un risultato scadente, ho pensato di ingrandire il set usato dal COM-MODORE ricavando i dati dalla ROM generatrice di caratteri, conseguendo un risultato veramente buono, e risparmiando una quantità enorme di memoria. Vediamo i dati tecnici della routine stessa: Come prima cosa, per far sì che la routine funzioni correttamente bisogna definire alcuni parametri essenziali alla stessa, che sono:

A\$ = (Altezza lettera) va da un minimo di 1 a un massimo di 255

L\$ = (Larghezza lettera) idem come sopra

X\$ = (Calcola asse X) va da 0 a 255

Y\$ = (Calcola asse Y) va da 0 a 255

P\$ = (Stringa da ingrandire)

Esempio:

A\$ = "5": L\$ = "3": X\$ = "30":

Y\$ = "100": P\$ = "VINCENZO"

Con questi parametri abbiamo indicato alla routine che la stringa VINCENZO dovrà essere ingrandita in altezza per 5, in larghezza per 3, e dovrà essere stampata a 30 pixel partendo dal margine sinistro e di 100 pixel dall'alto.

Questa routine lavora in alta risoluzione (hires), e vediamo per sommi capi come:

Per prima cosa verrà attivato lo schermo in alta risoluzione, quindi verrà ripulito (non abbiate timore di dover stare davanti allo schermo una decina di minuti, perché in l/m pulire 8192 bytes significa impiegare circa 9 decimi di secondo e con una buona ed accorta programmazione sicuramente molto meno); il programma quindi passerà a ricercare in memoria i parametri di A\$,



L\$, X\$, Y\$, e dopo averli trovati li immagazzinerà per poterli poi utilizzare; quindi passerà a trovare la stringa da ingrandire ed una volta trovata, caricherà il primo dato e lo ricercherà nella ROM generatrice di caratteri.

Vediamo come far funzionare la routine: Dando SYS 49661 si entra nella routine cancellando quello che si trova in pagina grafica, se invece volete rientrare senza cancellare la vecchia scritta o anche per aggiungere qualche altra cosa date SYS 49664. Una volta ritornato al BASIC e data la pausa voluta, basta dare SYS 50120 per disattivare la pagina grafica ma senza cancellare la scritta anche se essa non sarà più vi-

sibile: vengono impostate da capo le aree di memoria per lo schermo e per il video e viene restituito il controllo al BASIC. Se volete cambiare il colore alla scritta o al fondo fate una POKE alla locazione 50101 ricordando che i 4 bit più bassi indicano il colore di fondo mentre quelli alti indicano il colore di tutti i bit impostati ad uno.

Per battere questo programma usate il caricatore di l/m "MLX", pubblicato in altra parte della rivista. Alle sue domande circa indirizzo di partenza e di fine bisognerà rispondere rispettivamente con 49152 e 50345. Il programma si carica con: LOAD "INGRANDIRE", 1,1.

49152 :032,041,197,169,000,141,068
 49158 :032,208,169,011,141,033,088
 49164 :208,169,060,141,132,003,213
 49170 :169,147,032,210,255,173,236
 49176 :030,208,169,000,141,120,180
 49182 :003,169,144,032,210,255,075
 49188 :160,000,185,071,201,201,086
 49194 :008,240,007,153,000,050,244
 49200 :200,076,038,192,169,138,093
 49206 :133,252,169,197,133,253,167
 49212 :160,000,177,252,201,000,082
 49218 :240,012,032,210,255,230,021
 49224 :252,208,243,230,253,076,054
 49230 :062,192,169,063,141,021,214
 49236 :208,169,200,141,248,007,033
 49242 :141,250,007,141,251,007,119
 49248 :141,252,007,141,253,007,129
 49254 :169,201,141,249,007,169,014
 49260 :000,162,000,157,000,208,123
 49266 :232,224,015,208,248,169,186
 49272 :015,141,039,208,169,146,070
 49278 :141,000,208,169,141,141,158
 49284 :001,208,169,140,141,002,025
 49290 :208,169,135,141,003,208,234
 49296 :169,002,141,040,208,169,105
 49302 :004,141,041,208,169,014,215
 49308 :141,042,208,169,007,141,096
 49314 :043,208,169,013,141,044,012
 49320 :208,169,000,141,016,208,142
 49326 :169,255,141,062,003,032,068
 49332 :011,194,169,255,141,015,197
 49338 :212,169,128,141,018,212,042
 49344 :173,030,208,169,049,032,085
 49350 :136,196,169,254,141,066,136
 49356 :003,169,255,141,067,003,074
 49362 :032,154,196,169,017,141,151
 49368 :005,212,169,243,141,006,224
 49374 :212,169,033,141,004,212,225
 49380 :032,249,192,032,139,193,041
 49386 :032,249,192,032,193,194,102
 49392 :032,064,196,032,015,197,008
 49398 :076,228,192,141,060,003,178
 49404 :142,061,003,173,000,220,083
 49410 :041,008,208,039,174,000,216
 49416 :208,232,224,000,208,014,126
 49422 :173,016,208,009,001,141,050
 49428 :016,208,141,137,197,076,027
 49434 :042,193,173,016,208,041,187
 49440 :001,201,001,208,005,224,160
 49446 :009,208,001,202,142,000,088
 49452 :208,173,000,220,041,004,178
 49458 :208,039,174,000,208,202,113
 49464 :224,000,208,014,173,016,179
 49470 :208,041,254,141,016,208,162
 49476 :141,137,197,076,088,193,132
 49482 :173,016,208,041,001,201,202
 49488 :000,208,005,224,026,208,239
 49494 :001,232,142,000,208,173,074
 49500 :000,220,041,001,208,012,062
 49506 :174,001,208,202,224,054,193

49512 :208,001,232,142,001,208,128
 49518 :173,000,220,041,002,208,242
 49524 :012,174,001,208,232,224,199
 49530 :228,208,001,202,142,001,136
 49536 :208,173,060,003,174,061,039
 49542 :003,032,060,195,096,173,181
 49548 :062,003,041,003,168,173,078
 49554 :129,197,201,000,208,033,146
 49560 :192,000,240,111,173,130,230
 49566 :197,201,000,208,022,192,210
 49572 :001,240,100,173,131,197,238
 49578 :201,000,208,011,192,002,016
 49584 :240,089,173,132,197,201,184
 49590 :000,240,082,185,129,197,247
 49596 :201,000,240,069,170,192,036
 49602 :001,208,032,173,137,197,174
 49608 :041,008,201,000,240,023,201
 49614 :138,056,233,001,201,000,067
 49620 :208,027,173,137,197,041,227
 49626 :195,141,137,197,138,056,058
 49632 :233,002,076,241,193,138,083
 49638 :201,146,144,005,233,001,192
 49644 :076,241,193,105,001,153,237
 49650 :129,197,185,133,197,201,004
 49656 :141,144,005,233,001,076,080
 49662 :002,194,105,001,153,133,074
 49668 :197,136,192,255,208,175,143
 49674 :096,238,062,003,173,062,132
 49680 :003,206,132,003,041,003,148
 49686 :141,063,003,170,160,000,047
 49692 :169,000,141,064,003,238,131
 49698 :063,003,206,064,003,153,014
 49704 :129,197,200,192,004,208,202
 49710 :248,169,028,141,129,197,190
 49716 :169,192,141,137,197,173,037
 49722 :027,212,041,127,105,044,102
 49728 :141,133,197,224,000,240,231
 49734 :069,169,008,141,130,197,016
 49740 :169,008,013,137,197,141,229
 49746 :137,197,173,027,212,041,101
 49752 :127,105,044,141,134,197,068
 49758 :169,203,045,137,197,141,218
 49764 :137,197,224,001,240,034,165
 49770 :169,055,141,135,197,173,208
 49776 :027,212,041,127,105,067,179
 49782 :141,131,197,224,001,240,028
 49788 :015,169,227,141,136,197,241
 49794 :173,027,212,041,127,105,047
 49800 :067,141,132,197,224,000,129
 49806 :208,005,169,007,141,021,181
 49812 :208,224,001,208,005,169,195
 49818 :015,141,021,208,224,002,253
 49824 :208,005,169,031,141,021,223
 49830 :208,224,003,208,005,169,215
 49836 :063,141,021,208,238,063,138
 49842 :003,032,013,196,173,132,215
 49848 :003,201,014,240,003,238,115
 49854 :132,003,096,173,129,197,152
 49860 :201,000,240,012,141,004,026
 49866 :208,173,133,197,141,005,035

LOAD



SUPERVIC

49872 :208,076,222,194,169,000,053
 49878 :141,004,208,169,000,141,109
 49884 :005,208,173,130,197,201,110
 49890 :000,240,012,141,006,208,065
 49896 :173,134,197,141,007,208,068
 49902 :076,251,194,169,000,141,045
 49908 :006,208,169,030,141,007,037
 49914 :208,173,131,197,201,000,136
 49920 :240,012,141,008,208,173,014
 49926 :135,197,141,009,208,076,004
 49932 :024,195,169,000,141,008,037
 49938 :208,169,070,141,009,208,055
 49944 :173,132,197,201,000,240,199
 49950 :012,141,010,208,173,136,198
 49956 :197,141,011,208,076,053,210
 49962 :195,169,000,141,010,208,253
 49968 :169,111,141,011,208,173,093
 49974 :137,197,141,016,208,096,081
 49980 :173,030,208,141,065,003,168
 49986 :173,065,003,041,004,201,041
 49992 :004,208,022,169,000,141,104
 49998 :129,197,032,013,196,032,165
 50004 :217,196,173,021,208,041,172
 50010 :251,141,021,208,076,154,173
 50016 :196,173,065,003,041,008,070
 50022 :201,008,208,030,169,000,206
 50028 :141,130,197,173,137,197,059
 50034 :041,247,141,137,197,032,141
 50040 :013,196,032,238,196,173,200
 50046 :021,208,041,247,141,021,037
 50052 :208,076,154,196,173,065,236
 50058 :003,041,016,201,016,208,111
 50064 :022,169,000,141,131,197,036

50070 :032,013,196,032,249,196,100
 50076 :173,021,208,041,239,141,211
 50082 :021,208,076,154,196,173,222
 50088 :065,003,041,032,201,032,030
 50094 :208,022,169,000,141,132,078
 50100 :197,032,013,196,032,004,142
 50106 :197,173,021,208,041,223,025
 50112 :141,021,208,076,154,196,220
 50118 :096,169,015,141,024,212,087
 50124 :169,010,141,132,003,162,053
 50130 :255,142,001,212,202,142,140
 50136 :068,003,032,015,197,174,193
 50142 :068,003,224,000,208,239,196
 50148 :169,050,141,132,003,032,243
 50154 :015,197,238,032,208,173,073
 50160 :032,208,041,015,201,000,225
 50166 :208,241,169,000,141,024,005
 50172 :212,173,000,220,041,016,146
 50178 :208,249,104,104,104,104,107
 50184 :104,104,076,003,192,169,144
 50190 :146,141,000,208,169,141,051
 50196 :141,001,208,162,007,160,187
 50202 :035,024,032,240,255,206,050
 50208 :063,003,238,064,003,173,064
 50214 :063,003,024,105,048,032,057
 50220 :210,255,162,012,160,035,110
 50226 :024,032,240,255,173,064,070
 50232 :003,024,105,048,032,210,222
 50238 :255,096,165,197,201,004,212
 50244 :208,011,169,060,141,132,021
 50250 :003,169,049,032,136,196,147
 50256 :096,201,005,208,011,169,002
 50262 :042,141,132,003,169,050,111
 50268 :032,136,196,096,201,006,247
 50274 :208,011,169,035,141,132,026
 50280 :003,169,051,032,136,196,179
 50286 :096,201,003,208,011,169,030
 50292 :027,141,132,003,169,052,128
 50298 :032,136,196,096,173,141,128
 50304 :002,041,001,201,000,208,069
 50310 :247,096,141,082,003,162,097
 50316 :017,160,035,024,032,240,136
 50322 :255,173,082,003,032,210,133
 50328 :255,096,173,066,003,174,151
 50334 :067,003,024,105,002,144,247
 50340 :001,232,141,066,003,142,237

Leggete

COMPUTERWORLD ITALIA



di D. E. Smith
trad. e adatt. di
G. G. Barzaghi

Disegnatore vi aiuta a creare complessi disegni tridimensionali utilizzando il modo grafico ad alta risoluzione che il vostro computer mette a disposizione. Progettato in origine per il VIC 20 fornito di espansione di memoria da 16K, ne viene fornita anche la versione per C64.

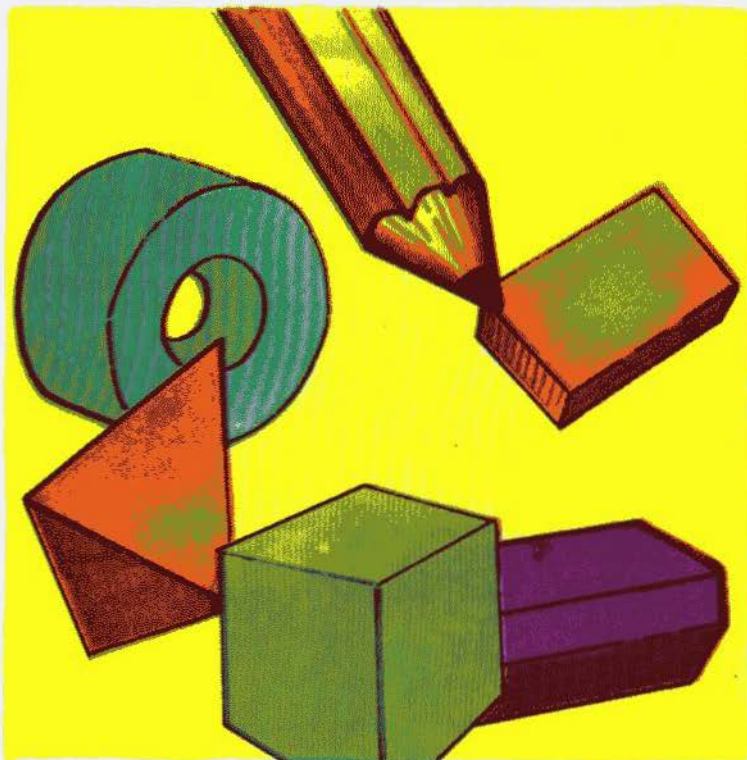
Disegnare con un joystick è divertente, ma manca della precisione e flessibilità che gli artisti ed i disegnatori pretendono da un attrezzo da disegno. Con "Disegnatore", potete creare complessi disegni sullo schermo. Limitatevi a fornire i punti iniziali e finali di ogni linea, e il Disegnatore tratterà la linea per voi.

Potete inoltre definire il punto in cui due linee partono ed il punto in cui si incontrano, o il punto in cui una singola linea cambia direzione, oppure potete definire i punti di una linea curva. Il Disegnatore vi consente anche di definire figure sia aperte che chiuse a due dimensioni così come disegni complessi che utilizzano delle combinazioni di figure a due e a tre dimensioni

Coordinate cartesiane

Disegnatore sfrutta il modo grafico ad alta risoluzione del vostro calcolatore, permettendovi così di disegnare controllando i singoli pixel (punti luminosi) dello schermo. In alta risoluzione, potrete localizzare la posizione di un singolo pixel usando un sistema di coordinate cartesiane, il diffusissimo sistema di coordinate X-Y utilizzato nel tracciamento di grafici e disegni. La ascissa X individua punti sull'asse orizzontale, e l'ordinata Y su di un asse verticale. Potete localizzare qualsiasi punto sullo schermo dandone le distanze, misurate lungo l'asse orizzontale e l'asse verticale, da un punto assunto come origine degli assi. Esattamente nello stesso modo in cui potete individuare una certa via sulla piantina di una città, cercandola nel riquadro chiamato F-5 o C-2.

I punti dell'ascissa X cominciano al margine sinistro dello schermo e si incrementano verso destra, mentre le or-



dinate Y iniziano al margine superiore dello schermo e si incrementano verso il basso. La posizione di coordinate (0,0) coincide con la posizione HOME del cursore, nell'angolo superiore sinistro dello schermo. Quindi, la posizione di coordinate (36,45) è ad ascissa 36 e ad ordinata 45 rispetto alla posizione di coordinate (0,0). Lo schermo ad alta risoluzione del VIC ha una ampiezza di 175 punti lungo l'asse X e di 160 lungo l'asse Y, per un totale di 28000 punti singolarmente localizzabili sullo schermo. (Il 64 ha 320 punti lungo l'asse X e 200 lungo l'Y, per un totale di 64000 punti). Per mostrarvi come il Disegnatore utilizza questo sistema di tracciamento, creiamo passo per passo il disegno di

Figura 1.

Innanzitutto, copiate il programma e registratelo. Quindi, spegnete il VIC e riccendetelo. Prima di caricare il programma, dovete spostare l'inizio del BASIC alla locazione di memoria 8192 (per sistemare l'espansione di 16K) inserendo la riga seguente, in modo diretto da tastiera:

POKE44,32:POKE8192,0:NEW

ciò va fatto ogni volta che caricate il programma.) (La versione per il C64 comprende i programmi 1 e 2. Copiate separatamente ciascuno di essi e registrateli su nastro o disco. Caricate e mandate in esecuzione il programma 1 prima di man-

dare in esecuzione il programma 2.

Il Disegnatore in funzione

Ora caricate il programma e mandatelo in esecuzione.

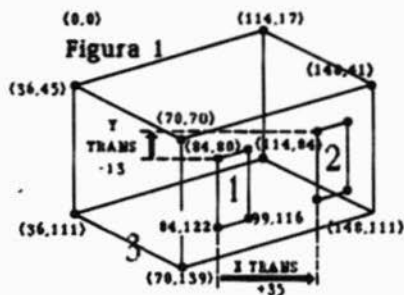
Il Disegnatore comincia con il mostrarvi le sette opzioni del suo menu:

- D - Definizione
- L - Lettura oggetto
- R - Registrazione oggetto
- V - Visualizzazione oggetto
- T - Traslazione oggetto
- M - Menu principale
- F - Fine

Scegliete D per definire l'oggetto che volete disegnare sullo schermo.

In questa seduta di prova, definirete le diverse figure contenute in Figura 1, un parallelepipedo con due finestre nella parte anteriore. Se suddividete il disegno nei suoi componenti, ottenete una parte tridimensionale, il parallelepipedo, e due parti bidimensionali, le due finestre.

Premendo D comincia una serie di quesiti che vi chiedono di fornire i dati necessari per disegnare l'oggetto. Il primo quesito riguarda il numero di parti bidimensionali (cioè a due dimensioni), a cui potete pensare come una serie di superfici piane da rappresentare sullo schermo. In Figura 1, la scatola a forma di parallelepipedo è disegnata in modo da simulare tre dimensioni, ma le due finestre vengono disegnate come rettangoli piani, quindi rispondete 2 per quanto riguarda il numero di parti in due dimensioni.



Queste parti bidimensionali sono definite dalla successiva serie di quesiti. Vi si chiede il numero dei punti che costituiscono la prima parte del disegno. La Finestra n. 1 è composta da quattro punti, quindi rispondete 4. Il Disegnatore quindi vi richiede le coordinate X,Y di ogni punto. (Ricordate il nostro sistema di coordinate X,Y). Per congruenza e correttezza, inserite le coordinate in ordine orario. Assicuratevi di inserire ogni gruppo di coordinate correttamente prima di premere RETURN. Il Disegnatore non ha possibilità di correzione, ed un errore può provocare l'uscita di figure malfatte. Ogni volta che premete RETURN, vi verranno chie-

ste le coordinate del punto successivo.

- X,Y (Inserite) 84,80
- X,Y (Inserite) 99,73
- X,Y (Inserite) 99,116
- X,Y (Inserite) 84,122

Dal momento che avete due finestre (due parti bidimensionali), il quesito successivo vi chiede i dati della seconda finestra. Il Disegnatore vi permette di ottenere una seconda finestra, dai dati già inseriti per la prima.

Volete copiare l'ultima parte? (s/n) Rispondete s, perché volete che entrambe le finestre siano identiche, ma in posizioni diverse dello schermo. Il quesito successivo chiede l'entità della traslazione della figura sia in direzione X che in direzione Y.

Questa richiesta fissa a che distanza orizzontalmente (X) e verticalmente (Y) dalla sua posizione originale volete copiare la prima finestra per generare la seconda.

Diamo un altro sguardo alla Figura 1. Relativamente all'origine (0,0), la Finestra 2 è spostata a destra e verso l'alto rispetto alla Finestra 1. Per rendere l'esempio il più chiaro possibile, abbiamo già calcolato le distanze: La Finestra 2 va traslata di 35 unità lungo l'asse X verso destra, e di 13 lungo l'asse Y verso l'alto, rispetto alla Finestra 1. Il Disegnatore vi suggerisce:

PARTE2/TRASLAZIONE:

DESTRA = + X SINISTRA = - X
BASSO = + Y ALTO = - Y

Questo vi dice che le traslazioni lungo l'asse X vengono considerate positive se state spostando la figura verso destra, e negative se verso sinistra. Le traslazioni lungo l'asse Y sono positive se verso il basso e negative se state traslando verso l'alto rispetto alla posizione originale della figura traslata.

Vi verrà chiesto TRASLAZIONE X? e voi basterete 35. In risposta a TRASLAZIONE Y? risponderete -13, dato che le finestre non sono tridimensionali, rispondete con N alla domanda:

La parte è tridimensionale?

Non avete ancora creato il parallelepipedo che circonda le finestre.

Quando il calcolatore chiede:

Numero di parti in tre dimensioni? rispondete, ovviamente, 1, per il parallelepipedo.

Potete ora creare l'impressione della tridimensionalità, rappresentando certi spigoli del parallelepipedo e lasciando che il Disegnatore completi il parallelepipedo al vostro posto. Vedere, al riguardo la Figura 2.

Questo disegno rappresenta un parallelepipedo analogo a quello che siete sul punto di creare (Le linee tratteggiate sarebbero nascoste se il parallelepipedo fosse solido). Questo parallelepipedo ha sei lati, o facce: anteriore (Faccia 1), posteriore (Faccia 2), superiore (Faccia 3), inferiore (Faccia 4), ed i lati sinistro e destro, che, per evitare confusione, non sono etichettati. Ogni fac-

Figura 2 a

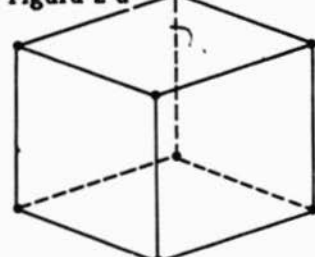


Figura 2 b

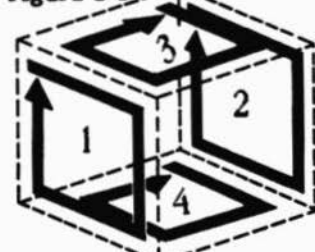


Figura 2 c



cia rappresenta una figura chiusa, a quattro lati. (Vedi Figure 2b e 2c).

Innanzitutto vi si chiede il numero di punti su ciascuna faccia:

Numero di punti su ciascuna faccia? 4
Figura chiusa? S

Ora inserite i punti di due lati opposti.

Faccia 1

- X,Y?36,45
- X,Y?70,70
- X,Y?70,139
- X,Y?36,111

Faccia 2

- X,Y?114,17
- X,Y?148,41
- X,Y?148,111
- X,Y?114,84

Dopo che avete inserito la quarta serie di coordinate e premuto RETURN, rientrate nel menu principale. Da questo menu, voi potete visualizzare il vostro disegno premendo V. Dopo averlo visto, premete M per rientrare nel menu principale. Potete registrare i vostri disegni sia su disco che su nastro, e ricaricarli in memoria usando l'opzione L del menu principale.

Traslazione

Una ulteriore opzione del Disegnatore

è la possibilità di muovere un intero oggetto entro lo schermo. Questa opzione di traslazione lavora come l'opzione interna al programma che permette di copiare l'ultima figura creata. L'opzione vi richiede i valori numerici delle traslazioni lungo gli assi X ed Y. Può essere efficacemente utilizzata per centrare un oggetto o per altri scopi. Si limita a cambiare, fisicamente, i dati di base. Se non volete conservare la traslazione dopo averla vista, non registratela.

Il programma permette di vedere un oggetto piazzato metà dentro e metà fuori dallo schermo. Disegnatore dispone di un sistema di sicurezza che non permette l'impiego di istruzioni di POKE al di fuori dell'area dello schermo. Il tracciatore continuerà a calcolare quei punti, ma non verranno attivati dalle istruzioni POKE.

Alcuni suggerimenti sulle rappresentazioni

Può darsi che vi siate accorti che, dopo aver premuto il tasto V, c'è una pausa e, magari, avete visto delle figure casuali mentre le locazioni dello schermo vengono cancellate. Ciò è normale. Anche quando non si vedono "rifiuti", il procedimento di cancellazione è in atto. Pazientate, e, ben presto, la vostra figura comincerà a formarsi. Quando premete M per rientrare nel menu principale, l'immagine viene cancellata nello stesso modo.

Quando il vostro disegno compare, potrete notare che le linee verticali sono più larghe delle orizzontali. Ciò ha a che fare con il modo in cui il tracciatore funziona. L'ampiezza di un carattere è normalmente di 8 bit per 8 bit. Per poter aumentare le dimensioni dell'area su cui si può disegnare, senza una eccessiva spesa in termini di occupazione di memoria, le dimensioni di un carattere sono state cambiate in 8 bit di larghezza per 16 bit di altezza.

Vi potrà lasciare un po' perplessi il modo in cui il Disegnatore traccia le linee diagonali. Invece di disegnare una linea retta, disegna una specie di piccola scaletta tra due punti. Ciò rappresenta una diretta conseguenza della ri-

soluzione del chip VIC-II. Qualsiasi sistema a raggio di scansione, non importa quanto precisa sia la sua risoluzione, produrrà, suppergiù, gli stessi effetti. I gradini si notano meno se disegnate, quanto più è possibile, in grande. Cercate di utilizzare tutto lo schermo.

Linee nascoste nelle rappresentazioni tridimensionali

Nella rappresentazione tridimensionale, l'oggetto è disegnato come se fosse trasparente: vale a dire, potete vedere tutti gli spigoli.

Ciò rappresenta una fonte di confusione. Il Disegnatore non possiede nessun algoritmo in grado di rimuovere automaticamente le linee nascoste, ma sotto certe condizioni potete aggirare i problemi della rappresentazione tridimensionale, disegnando in modo da simularla. La Figura 3a mostra un disegno a linee nascoste rappresentante un solido a forma di parallelepipedo in

prospettiva. Vengono mostrati solo gli spigoli in vista. Potete disegnarlo in due dimensioni come due parti (due figure rettangolari). Oppure possiamo farlo disegnare alla rappresentazione tridimensionale come se fosse una sola parte. Qui sopra il segreto. Niente ci impedisce di usare lo stesso punto due volte.

La prima faccia viene inserita normalmente: a cominciare dal punto a, quindi b, c e d. La seconda faccia viene inserita in maniera differente: punti e, f, b e a. Se fate attenzione al modo in cui il chip VIC-II traccia le linee, vedrete come disegni la stessa linea più di una volta. Ma il risultato è la figura voluta (Figura 3b).

Una tavola grafica

Per poter definire con precisione i vostri oggetti, relativamente ai punti che li costituiscono, è necessario costruirsi una tavoletta grafica. Tutto ciò di cui avete bisogno è una matita, un righello ed un foglio di carta, approssimativamente, di 24 centimetri per 40. Sul foglio, da sinistra a destra, tracciate 18 linee verticali a due centimetri una dall'altra. Dal margine superiore a quello inferiore, tracciate 17 linee orizzontali distanti un centimetro e mezzo. Numerate le linee verticali da 0 a 170 di dieci in dieci (0, 10, 20...) e quelle orizzontali da 0 a 160, sempre di dieci in dieci. Siate il più possibile precisi. Ora suddividete ogni riquadro in dieci sezioni identiche (con nove linee) orizzontalmente e verticalmente. Noterete che le linee verticali terminano a 170. Dovete aggiungere sei piccole sezioni al di là dell'ultima linea per arrivare a 175: la posizione più estrema a destra (coordinate X) dello schermo.

Ora avete una tavola grafica da usare con il Disegnatore. L'oggetto o il disegno che volete riprodurre sullo schermo dovrebbe essere disegnato o tracciato su di un foglio di carta da lucido e appoggiato alla tavola grafica. Quindi prendete nota sul disegno delle coordinate X e Y di ogni punto significativo. In notazione standard l'ascissa X viene fornita per prima, seguita dalla ordinata Y, separate da virgola (ax, ay).

Figura 3 a

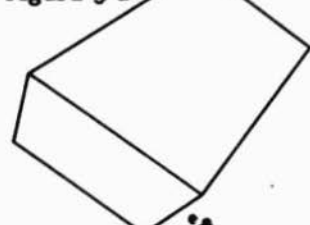
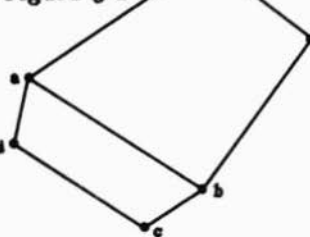


Figura 3 b



Disegnatore versione per C64

Programma 1

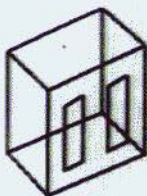
```
100 POKE16384,0:POKE16385,0      :rem 238
145 POKE641,0:POKE642,64          :rem 91
150 POKE43,1:POKE44,64:POKE55,0:POKE56,12
    8:POKE646,1:PRINT{CLR}        :rem 190
```

Programma 2

```
0 POKE53280,1                      :rem 190
1020 DIMA(20,2,26)                 :rem 131
1030 QS=""                          :rem 186
1040 REM MENU                       :rem 222
1050 PRINT{CLR}                     :rem 42
1060 PRINTSPC(14)"DISEGNATORE":PRINT:PRINT
    T:PRINT                          :rem 190
```

```
1070 PRINTSPC(7)"D-DEFINIZIONE OGGETTO":P
    RINT                             :rem 140
1080 PRINTSPC(7)"L-LETTURA OGGETTO":PRINT
    :rem 130
1090 PRINTSPC(7)"R-REGISTRAZIONE OGGETTO"
    :PRINT                           :rem 78
1100 PRINTSPC(7)"V-VISUALIZZAZIONE OGGETT
    O":PRINT                         :rem 251
1110 PRINTSPC(7)"T-TRASLAZIONE OGGETTO":P
    RINT                             :rem 175
1120 PRINTSPC(7)"M-MENU' PRINCIPALE":PRINT
    T                               :rem 135
1130 PRINTSPC(7)"F-FINE"            :rem 153
1140 GOSUB3700                      :rem 16
1150 GP=- (QS="D")-2*(QS="L")-3*(QS="R")-4
    *(QS="V")-5*(QS="T")           :rem 225
```

LOAD



```

1155 ONGPGOTO1190,3000,3250,1670,2700      :rem 216
1160 IFQS="F"THEN3740                        :rem 144
1170 GOTO1050                                :rem 200
1180 REM DEFINIZIONE                          :rem 226
1190 PRINT"{CLR}"                             :rem 47
1200 REM IN DUE DIMENSIONI                   :rem 11
1210 PRINT"AL MASSIMO:"PRINT"20 PARTI":P      :rem 80
    RINT"12 PUNTI PER PARTE/FACCIA":PRIN
    T
1220 PRINT"NUMERO DI PARTI IN DUE DIMENSI     :rem 40
    ONI":INPUTP1
1230 PA=P1+2:QS="":A(0,0,0)=P1              :rem 20
1240 IFP1=0THEN1410                           :rem 57
1250 FORPT=2TOPT+1                             :rem 66
1260 PRINT:PRINT"PARTE";PT-1                  :rem 26
1270 IFPT=2THEN1320                           :rem 97
1280 PRINT"VUOI COPIARE L'ULTIMA PARTE? (    :rem 129
    S/N)":GOSUB3700
1290 IFQS="N"THEN1320                           :rem 148
1300 GOSUB2430                                :rem 13
1310 GOTO1390                                :rem 203
1320 INPUT"NUMERO DI PUNTI";PO                :rem 103
1330 PRINT"FIGURA CHIUSA? (S/N)":GOSUB370    :rem 247
    0
1340 IFQS="S"THENA(PT,0,1)=1:GOTO1360        :rem 105
1350 A(PT,0,1)=2                              :rem 39
1360 A(PT,0,0)=PO                             :rem 148
1370 TA=2                                     :rem 207
1380 FORT=2TOPO+1:INPUT"X,Y";A(PT,0,TA),A    :rem 223
    (PT,0,TA+1):TA=TA+2:NEXT
1390 NEXTPT                                   :rem 176
1400 REM IN TRE DIMENSIONI                    :rem 26
1410 PRINT"NUMERO DI PARTI IN TRE DIMENSI     :rem 55
    ONI":INPUTP2
1420 IFP2=0THENA(1,0,0)=0:GOTO1040           :rem 148
1430 QS="":A(1,0,0)=P2                        :rem 50
1440 FORPT=PATOP2+PA-1                        :rem 97

```

```

1450 PRINT:PRINT"PARTE";PT-1                 :rem 27
1460 IFPT=PATHEN1510                          :rem 194
1470 PRINT"VOLETE COPIARE L'ULTIMA PARTE?    :rem 14
    (S/N)":GOSUB3700
1480 IFQS="N"THEN1510                          :rem 150
1490 GOSUB2430                                :rem 23
1500 GOTO1630                                :rem 201
1510 PRINT"NUMERO DI PUNTI SU OGNI FACCIA    :rem 54
    ?"
1520 INPUTPO                                  :rem 247
1530 A(PT,0,0)=PO                             :rem 147
1540 PRINT"FIGURA CHIUSA? (S/N)":GOSUB370    :rem 250
    0
1550 IFQS="S"THENA(PT,0,1)=1:GOTO1570        :rem 111
1560 A(PT,0,1)=2                              :rem 42
1570 TA=2                                     :rem 209
1580 FORT=1TO2                                :rem 160
1590 PRINT"FACCIA";TT                         :rem 26
1600 FORT=2TOPO+1:INPUT"X,Y";A(PT,TT,TA),    :rem 202
    A(PT,TT,TA+1):TA=TA+2:NEXT
1610 TA=2                                     :rem 204
1620 NEXTTT                                   :rem 176
1630 NEXTPT                                   :rem 173
1640 GOTO1050                                :rem 202
1650 REM                                       :rem 176
1670 POKE53272,(PEEK(53272)OR8):POKE53265    :rem 234
    ,PEEK(53265)OR32
1700 FORI=0TO999:POKE1024+I,1:NEXT           :rem 45
1710 CO=0                                     :rem 200
1730 FORI=8192TO8192+7999:POKEI,0:NEXT      :rem 23
1740 REM LETTURA IN DUE DIMENSIONI          :rem 53
1750 P1=A(0,0,0):PA=P1+2                     :rem 235
1760 IFP1=0THEN1900                           :rem 68
1770 FORPT=2TOPT+1                             :rem 73
1780 PO=A(PT,0,0)                             :rem 154
1790 TA=2                                     :rem 213
1800 FORT=2TOPO+1                             :rem 17
1810 X1=A(PT,0,TA):Y1=A(PT,0,TA+1)           :rem 147
1820 REM CONFRONTO CON LA PRIMA VISUALIZZ    :rem 228
    AZIONE
1830 IFT=PO+1ANDA(PT,0,1)=1THENX2=A(PT,0,    :rem 115
    2):Y2=A(PT,0,3):GOSUB2160
1840 IFT=PO+1THEN1890                           :rem 233
1850 X2=A(PT,0,TA+2):Y2=A(PT,0,TA+3)         :rem 248
1860 GOSUB2160                                :rem 24
1870 TA=TA+2                                  :rem 148
1880 NEXTT                                    :rem 100
1890 NEXTPT                                   :rem 181
1900 REM LETTURA IN TRE DIMENSIONI          :rem 64
1910 P2=A(1,0,0)                              :rem 5
1920 IFP2=0THEN2340                           :rem 66
1930 FORPT=PATOP2+PA-1                         :rem 101
1940 PO=A(PT,0,0)                             :rem 152
1950 FORT=1TO2                                :rem 161
1960 TA=2                                     :rem 212
1970 FORT=2TOPO+1                             :rem 25
1980 X1=A(PT,TT,TA):Y1=A(PT,TT,TA+1)         :rem 139
1990 IFT=PO+1ANDA(PT,0,1)=1THENX2=A(PT,TT    :rem 106
    ,2):Y2=A(PT,TT,3):GOSUB2160
2000 IFT=PO+1THEN2050                           :rem 211
2010 X2=A(PT,TT,TA+2):Y2=A(PT,TT,TA+3)       :rem 221
2020 GOSUB2160                                :rem 13
2030 TA=TA+2                                  :rem 137

```

2040	NEXTT	:rem	89	2580	RETURN	:rem	175
2050	NEXTTT	:rem	174	2590	A(PT,0,0)=A(PT-1,0,0):A(PT,0,1)=A(PT	:rem	250
2060	TA=2	:rem	204	-1,0,1)	:rem	154	
2070	FORFA=1TOPO	:rem	231	2600	FORTT=1TO2	:rem	205
2080	X1=A(PT,1,TA):Y1=A(PT,1,TA+1)	:rem	149	2610	TA=2	:rem	97
2090	X2=A(PT,2,TA):Y2=A(PT,2,TA+1)	:rem	154	2620	FORT=2TOA(PT,0,0)+1	:rem	211
2100	GOSUB2160	:rem	12	2630	A(PT,TT,TA)=A(PT-1,TT,TA)+TX	:rem	141
2110	TA=TA+2	:rem	136	2640	A(PT,TT,TA+1)=A(PT-1,TT,TA+1)+TY	:rem	145
2120	NEXTFA	:rem	139	2650	TA=TA+2	:rem	97
2130	NEXTPT	:rem	169	2660	NEXTT	:rem	182
2140	GOTO2340	:rem	201	2670	NEXTTT	:rem	52
2150	REM CONFRONTO CON LA SECONDA VISUALI			2680	PRINT"{CLR}"	:rem	177
2160	ZZAZIONE	:rem	101	2690	RETURN	:rem	249
2170	X=0:Y=0:SI=1	:rem	210	2700	REM TRASLAZIONE	:rem	191
2180	IFX1>X2ANDY1>Y2THENSI=-1	:rem	54	2710	PRINT"{CLR}":PRINT"TRASLAZIONE:"	:rem	78
2190	IF(X2-X1)<(Y2-Y1)ANDY1<Y2THEN2200	:rem	209	2720	PRINT"DESTRA=+X SINISTRA=-X"	:rem	144
2200	GOTO2220	:rem	203	2730	PRINT"BASSO =+Y ALTO(4 SPAZI)=-Y":P	:rem	44
2210	IF(Y2-Y1)>(X1-X2)THEN2300	:rem	169	RINT	:rem	47	
2220	SI=-1:GOTO2230	:rem	54	2740	INPUT"TRASLAZIONE X ";TX	:rem	237
2230	IF(Y1-Y2)<(X1-X2)ANDY1>Y2THEN2280	:rem	214	2750	INPUT"TRASLAZIONE Y ";TY	:rem	74
2240	IFY1=Y2ANDX1=X2THENRETURN	:rem	218	2760	P1=A(0,0,0):PA=P1+2	:rem	155
2250	IFY1=Y2THEN3620	:rem	163	2770	FORPT=2TOP1+1	:rem	214
2260	REM CICLO DI TRACCIAMENTO DELLA VISU			2780	PO=A(PT,0,0)	:rem	18
2270	ALIZZAZIONE	:rem	23	2790	TA=2	:rem	133
2280	FORYY=Y1TOY2STEPY1*((Y2-Y1)/(X2-X1))	:rem	20	2800	FORT=2TOPO+1	:rem	63
2290	:Y=INT(YY):X=X1	:rem	202	2810	A(PT,0,TA)=A(PT,0,TA)+TX	:rem	145
2300	GOSUB3500:GOSUB3560:NEXTYY	:rem	203	2820	A(PT,0,TA+1)=A(PT,0,TA+1)+TY	:rem	97
2310	GOTO2320	:rem	203	2830	TA=TA+2	:rem	178
2320	IF(Y1-Y2)>(X1-X2)THENSI=-1:GOTO2300	:rem	91	2840	NEXTT	:rem	10
2330	GOTO2230	:rem	205	2850	NEXTPT	:rem	105
2340	IFX1=X2THEN3660	:rem	162	2860	P2=A(1,0,0)	:rem	156
2350	FORXX=X1TOX2STEPX1*((X2-X1)/(Y2-Y1))	:rem	11	2870	FORPT=PAOP2+PA-1	:rem	165
2360	:X=INT(XX):Y=Y1	:rem	199	2880	PO=A(PT,0,0)	:rem	207
2370	GOSUB3500:GOSUB3590:NEXTXX	:rem	167	2890	FORTT=1TO2	:rem	20
2380	RETURN	:rem	141	2900	TA=2	:rem	119
2390	REM RITORNA AL MENU	:rem	175	2910	FORT=2TOPO+1	:rem	49
2400	AS=""	:rem	183	2920	A(PT,TT,TA)=A(PT,TT,TA)+TX	:rem	147
2410	GETAS:IFAS=""THEN2350	:rem	242	2930	A(PT,TT,TA+1)=A(PT,TT,TA+1)+TY	:rem	99
2420	IFAS<>CHR\$(77)THEN2350	:rem	24	2940	TA=TA+2	:rem	184
2430	FORI=8192TO8192+7999:POKEI,0:NEXT	:rem	145	2950	NEXTT	:rem	181
2440	POKE53272,21:POKE53265,27	:rem	73	2960	NEXTTT	:rem	210
2450	PRINTCHR\$(147)	:rem	197	2970	NEXTPT	:rem	217
2460	GOTO1050	:rem	158	2980	GOTO1050	:rem	117
2470	END	:rem	120	2990	REM LETTURA	:rem	100
2480	REM COPIA UNA PARTE	:rem	37	3000	PRINT"{CLR}NOME DISTINTIVO DEL DISEG	:rem	185
2490	PRINT"{CLR}":PRINT"PARTE";PT-1;"/TRA	:rem	77	NO":INPUTFS	:rem	103	
2500	SLAZIONE:"	:rem	143	3001	PRINT"{CLR}NASTRO O DISCO (N/D)":INP	:rem	228
2510	PRINT"DESTRA=+X SINISTRA=-X"	:rem	156	UTOUS	:rem	234	
2520	PRINT"BASSO =+Y ALTO(4 SPAZI)=-Y":P	:rem	144	3003	IFOUS<>"N"ANDOUS<>"D"THEN3001	:rem	247
2530	RINT	:rem	43	3005	IFOUS="N"THENOU=1:GOTO3020	:rem	58
2540	INPUT"TRASLAZIONE X ";TX	:rem	46	3008	OU=8	:rem	204
2550	INPUT"TRASLAZIONE Y ";TY	:rem	95	3020	OPEN1,OU,0,FS	:rem	226
2560	PRINT"LA PARTE E' TRIDIMENSIONALE?(S	:rem	166	3030	INPUT#1,A(0,0,0),A(1,0,0)	:rem	96
2570	/N)":GOSUB3700	:rem	241	3040	FORPT=2TOA(0,0,0)+1	:rem	9
2580	IFQS="S"THEN2590	:rem	204	3050	TA=2	:rem	144
2590	A(PT,0,0)=A(PT-1,0,0):A(PT,0,1)=A(PT	:rem	226	3060	INPUT#1,A(PT,0,0),A(PT,0,1)	:rem	87
2600	-1,0,1)	:rem	96	3070	FORT=1TOA(PT,0,0)+1	:rem	168
2610	TA=2	:rem	226	3080	INPUT#1,A(PT,0,TA),A(PT,0,TA+1)	:rem	132
2620	FORT=2TOA(PT,0,0)+1	:rem	156	3090	TA=TA+2	:rem	224
2630	A(PT,TT,TA)=A(PT-1,TT,TA)+TX	:rem	144	3100	NEXTT	:rem	
2640	A(PT,TT,TA+1)=A(PT-1,TT,TA+1)+TY	:rem	96	3110	NEXTPT	:rem	
2650	TA=TA+2	:rem	144	3120	FORPT=A(0,0,0)+2TOA(1,0,0)+A(0,0,0)+	:rem	
2660	NEXTT	:rem	96	1	:rem	:rem	
2670	NEXTTT	:rem	132	3130	INPUT#1,A(PT,0,0),A(PT,0,1)	:rem	
2680	PRINT"{CLR}"	:rem	224				
2690	RETURN	:rem					
2700	REM TRASLAZIONE	:rem					
2710	PRINT"{CLR}":PRINT"TRASLAZIONE:"	:rem					
2720	PRINT"DESTRA=+X SINISTRA=-X"	:rem					
2730	PRINT"BASSO =+Y ALTO(4 SPAZI)=-Y":P	:rem					
2740	RINT	:rem					
2750	INPUT"TRASLAZIONE X ";TX	:rem					
2760	INPUT"TRASLAZIONE Y ";TY	:rem					
2770	P1=A(0,0,0):PA=P1+2	:rem					
2780	FORPT=2TOP1+1	:rem					
2790	PO=A(PT,0,0)	:rem					
2800	TA=2	:rem					
2810	FORT=2TOPO+1	:rem					
2820	A(PT,0,TA)=A(PT,0,TA)+TX	:rem					
2830	A(PT,0,TA+1)=A(PT,0,TA+1)+TY	:rem					
2840	TA=TA+2	:rem					
2850	NEXTT	:rem					
2860	NEXTPT	:rem					
2870	P2=A(1,0,0)	:rem					
2880	FORPT=PAOP2+PA-1	:rem					
2890	PO=A(PT,0,0)	:rem					
2900	FORTT=1TO2	:rem					
2910	TA=2	:rem					
2920	FORT=2TOPO+1	:rem					
2930	A(PT,TT,TA)=A(PT,TT,TA)+TX	:rem					
2940	A(PT,TT,TA+1)=A(PT,TT,TA+1)+TY	:rem					
2950	TA=TA+2	:rem					
2960	NEXTT	:rem					
2970	NEXTTT	:rem					
2980	NEXTPT	:rem					
2990	GOTO1050	:rem					
3000	REM LETTURA	:rem					
3001	PRINT"{CLR}NOME DISTINTIVO DEL DISEG	:rem					
3002	NO":INPUTFS	:rem					
3003	PRINT"{CLR}NASTRO O DISCO (N/D)":INP	:rem					
3004	UTOUS	:rem					
3005	IFOUS<>"N"ANDOUS<>"D"THEN3001	:rem					
3006	IFOUS="N"THENOU=1:GOTO3020	:rem					
3007	OU=8	:rem					
3008	OPEN1,OU,0,FS	:rem					
3009	INPUT#1,A(0,0,0),A(1,0,0)	:rem					
3010	FORPT=2TOA(0,0,0)+1	:rem					
3011	TA=2	:rem					
3012	INPUT#1,A(PT,0,0),A(PT,0,1)	:rem					
3013	FORT=1TOA(PT,0,0)+1	:rem					
3014	INPUT#1,A(PT,0,TA),A(PT,0,TA+1)	:rem					
3015	TA=TA+2	:rem					
3016	NEXTT	:rem					
3017	NEXTPT	:rem					
3018	FORPT=A(0,0,0)+2TOA(1,0,0)+A(0,0,0)+	:rem					
3019	1	:rem					
3020	INPUT#1,A(PT,0,0),A(PT,0,1)	:rem					



```

3140 FORTT=1TO2 :rem 154
3150 TA=2 :rem 205
3160 FORT=2TOA(PT,0,0)+1 :rem 97
3170 INPUT#1,A(PT,TT,TA),A(PT,TT,TA+1) :rem 249
3180 TA=TA+2 :rem 144
3190 NEXTT :rem 96
3200 NEXTTT :rem 172
3210 NEXTPT :rem 169
3220 CLOSE1 :rem 110
3230 GOTO1050 :rem 199
3240 REM FILE :rem 205
3250 PRINT"{CLR}" :rem 46
3260 PRINT"{CLR}NOME DISTINTIVO DEL DISEG :rem 125
NO":INPUTFS :rem 95
3261 PRINT"{CLR}NASTRO O DISCO (N/D)":INP :rem 175
UTINS :rem 92
3263 IFINS<>"N"ANDINS<>"D"THEN3261 :rem 223
IN=8 :rem 229
3270 OPEN1,IN,1,FS :rem 54
3280 RS=CHRS(13) :rem 188
3290 PRINT#1,A(0,0,0);RS;A(1,0,0) :rem 57
3300 FORPT=2TOA(0,0,0)+1 :rem 201
3310 TA=0 :rem 93
3320 FORT=0TOA(PT,0,0)+1 :rem 196
3330 PRINT#1,A(PT,0,TA);RS;A(PT,0,TA+1) :rem 142
3340 TA=TA+2 :rem 94
3350 NEXTT :rem 175
3360 NEXTPT :rem 139
3370 FORPT=A(0,0,0)+2TOA(1,0,0)+A(0,0,0)+ :rem 164
1 :rem 161
3380 PRINT#1,A(PT,0,0);RS;A(PT,0,1) :rem 203
3390 FORTT=1TO2 :rem 95
3400 TA=2 :rem 180
3410 FORT=2TOA(PT,0,0)+1 :rem 142
3420 PRINT#1,A(PT,TT,TA);RS;A(PT,TT,TA+1) :rem 94
3430 TA=TA+2 :rem 179
3440 NEXTT :rem 176
3450 NEXTTT :rem 117
3460 NEXTPT :rem 206
3470 CLOSE1 :rem 203
3480 GOTO1050 :rem 246
3490 REM SUBROUTINE DI TRACCIAMENTO :rem 7
3500 IFX>320ORX<0THENRETURN :rem 1
3510 IFY>199ORY<0THENRETURN :rem 243
3520 CH=INT(X/8):LN=YAND7 :rem 19
3530 RO=INT(Y/8):BY=8192+RO*320+8*CH+LN :rem 132
3540 BI=7-(XAND7):POKEBY,PEEK(BY)OR(2*BI) :rem 160
:RETURN :rem 150
3550 REM SUBROUTINE CICLO Y :rem 134
3560 IFX1<X2THENX1=X1+1:RETURN :rem 167
3570 X1=X1-1:RETURN :rem 132
3580 REM SUBROUTINE CICLO X :rem 134
3590 IFY1<Y2THENY1=Y1+1:RETURN :rem 167

```

```

3600 Y1=Y1-1:RETURN :rem 146
3610 REM Y1=Y2 :rem 0
3620 IFX1>X2THENS1=-1 :rem 17
3630 FORXX=X1TOX2STEPS1:X=INT(XX):Y=Y1 :rem 73
GOSUB3500:NEXTXX :rem 173
3640 RETURN :rem 2
3650 REM X1=X2 :rem 23
3660 IFY1>Y2THENS1=-1 :rem 84
3670 FORYY=Y1TOY2STEPS1:Y=INT(YY):X=X1 :rem 177
GOSUB3500:NEXTYY :rem 43
3680 RETURN :rem 192
3690 REM QS="" :rem 217
3700 QS="" :rem 172
3710 GETQS:IFQS=""THEN3710 :rem 211
3720 RETURN :rem 67
3730 REM FINE :rem 67
3740 PRINT"{CLR}":END :rem 67

```

Disegnatore versione per VIC 20

```

0 POKE36879,25 :rem 3
15 DIMA(20,2,26) :rem 38
20 QS="" :rem 88
49 REM MENU' :rem 173
50 PRINT"{CLR}{BLK}" :rem 89
51 PRINTSPC(3){SU}VIC-DISEGNATORE":PRINT :rem 4
:PRINT :rem 158
52 PRINTSPC(2)"D-DEFINIZIONE{ 11 SPAZI}OG :rem 38
GETTO":PRINT :rem 28
53 PRINTSPC(2)"L-LETTURA{ 15 SPAZI}OGGETT :rem 232
O":PRINT :rem 158
54 PRINTSPC(2)"R-REGISTRAZIONE{ 9 SPAZI} :rem 83
OGGETTO":PRINT :rem 176
55 PRINTSPC(2)"V-VISUALIZZAZIONE :rem 242
{ 7 SPAZI}OGGETTO":PRINT :rem 61
57 PRINTSPC(2)"T-TRASLAZIONE{ 11 SPAZI}OG :rem 138
GETTO":PRINT :rem 245
58 PRINTSPC(2)"M-MENU' PRINCIPALE":PRINT :rem 218
PRINT :rem 178
59 PRINTSPC(2)"F-FINE" :rem 213
60 GOSUB3700 :rem 18
65 ON-(QS="D")-2*(QS="L")-3*(QS="R")-4*(Q :rem 234
S="V")-5*(QS="T")GOTO100,1600,1800,300 :rem 254
,1399 :rem 194
68 IFQS="F"THEN3800 :rem 40
70 GOTO50 :rem 216
99 REM DEFINIZIONE :rem 103
100 PRINT"{CLR}" :rem 52
101 REM IN DUE DIMENSIONI :rem 370
102 PRINT"AL MASSIMO":PRINT"20 PARTI":PR :rem 200
INT"12 PUNTI PER OGNI PAR-TE/FACCIA" :rem 13
103 PRINT"NUMERO DI PARTI IN DUE DIMENSION :rem 247
I":INPUTP1 :rem 227
104 PA=P1+2:QS=""A(0,0,0)=P1 :rem 213
106 IFP1=0THEN200 :rem 18
107 FORPT=2TOPT+1 :rem 234
108 PRINT:PRINT"PARTE";PT-1 :rem 254
109 IFPT=2THEN120 :rem 194
110 PRINT"VOLETE COPIARE L'ULTI-MA PARTE :rem 40
(S/N)":GOSUB3700 :rem 216
111 IFQS="N"THEN120 :rem 103
115 GOSUB1000 :rem 52
117 GOTO140 :rem 370
120 INPUT"NUMERO DI PUNTI";PO :rem 200
125 PRINT"FIGURA CHIUSA ? (S/N)":GOSUB370 :rem 13
0 :rem 249
127 IFQS="S"THENA(PT,0,1)=1:GOTO129 :rem 102
:rem 152
128 A(PT,0,1)=2 :rem 102
129 A(PT,0,0)=PO :rem 152
130 TA=2 :rem 152

```

```

135  FORT=2*TOPO+1:INPUT"X,Y";A(PT,0,TA),A(PT,0,TA+1):TA=TA+2:NEXT      :rem 172
140  NEXTPT                      :rem 120
199  REM IN TRE DIMENSIONI      :rem 248
200  PRINT"NUMERO DI PARTI IN TREDIMENSIONI"
    I":INPUTP2                  :rem 3
202  IFP2=0THENA(1,0,0)=0:GOTO49 :rem 9
204  QS="":A(1,0,0)=P2         :rem 0
205  FORPT=PATOP2+PA-1         :rem 47
206  PRINT:PRINT"PARTE";PT-1   :rem 233
207  IFPT=PATHEN212           :rem 94
208  PRINT"VOLETE COPIARE L'ULTI-MA PARTE (S/N)":GOSUB3700 :rem 202
209  IFQS="N"THEN212           :rem 50
210  GOSUB1000                 :rem 212
211  GOTO229                   :rem 106
212  PRINT"NUMERO DI PUNTI "    :rem 89
213  PRINT"SU OGNI FACCIA";:INPUTPO :rem 119
    :rem 97
214  A(PT,0,0)=PO              :rem 97
216  PRINT"FIGURA CHIUSA ? (S/N)":GOSUB3700 :rem 201
    0 :rem 201
217  IFQS="S"THENA(PT,0,1)=1:GOTO220 :rem 5
    :rem 249
218  A(PT,0,1)=2              :rem 152
220  TA=2                      :rem 104
222  FORTT=1TO2               :rem 226
223  PRINT"FACCIA";TT         :rem 156
225  FORT=2*TOPO-1:INPUT"X,Y";A(PT,TT,TA),A(PT,TT,TA+1):TA=TA+2:NEXT :rem 158
226  TA=2                      :rem 130
227  NEXTTT                   :rem 128
229  NEXTPT                    :rem 65
298  GOTO50                   :rem 136
299  REM                       :rem 102
300  POKE36866,150:POKE36869,240:POKE648,3 :rem 172
    0 :rem 172
310  FORJ=217TO228:POKEJ,158:NEXT:FORJ=229 :rem 71
    TO250:POKEJ,159:NEXT :rem 71
320  POKE36865,30:POKE36867,21:POKE36869,2 :rem 172
    52:POKE36879,30 :rem 172
330  FORI=0TO219:POKE7680+I,I:NEXT :rem 18
    :rem 152
333  CO=0                      :rem 138
335  FORI=0TO219:POKE38400+I,CO:NEXT :rem 212
    :rem 9
340  FORI=4096TO7615:POKEI,0:NEXT:rem 212
349  REM LETTURA IN DUE DIMENSIONI :rem 9
    :rem 182
350  P1=A(0,0,0):PA=P1+2      :rem 230
351  IFP1=0THEN449             :rem 20
352  FORPT=2TOP1+1            :rem 101
353  PO=A(PT,0,0)              :rem 160
354  TA=2                      :rem 229
355  FORT=2*TOPO+1            :rem 229
360  X1=A(PT,0,TA):Y1=A(PT,0,TA+1) :rem 98
    :rem 190
399  REM CONFRONTO CON LA PRIMA VISUALIZZA :rem 12
    ZIONE :rem 124
405  IFT=PO+1ANDA(PT,0,1)=1THENX2=A(PT,0,2) :rem 192
    ):Y2=A(PT,0,3):GOSUB500 :rem 172
410  IFT=PO+1THEN445          :rem 96
420  X2=A(PT,0,TA+2):Y2=A(PT,0,TA+3) :rem 43
    :rem 128
430  GOSUB500                 :rem 23
435  TA=TA+2                  :rem 211
440  NEXTT                     :rem 73
445  NEXTPT                    :rem 73
449  REM LETTURA IN TRE DIMENSIONI :rem 73
    :rem 73
450  P2=A(1,0,0)              :rem 73
    :rem 73
451  IFP2=0THEN800            :rem 223
452  FORPT=PATOP2+PA-1        :rem 51
454  PO=A(PT,0,0)             :rem 103
458  FORTT=1TO2               :rem 115
459  TA=2                      :rem 166
460  FORT=2*TOPO+1            :rem 226
462  X1=A(PT,TT,TA):Y1=A(PT,TT,TA+1) :rem 85
    :rem 188
464  IFT=PO+1ANDA(PT,0,1)=1THENX2=A(PT,TT,2) :rem 176
    ):Y2=A(PT,TT,3):GOSUB500 :rem 97
466  IFT=PO+1THEN476          :rem 50
468  X2=A(PT,TT,TA+2):Y2=A(PT,TT,TA+3) :rem 136
    :rem 167
470  GOSUB500                 :rem 186
472  TA=TA+2                  :rem 105
474  NEXTT                     :rem 111
476  NEXTTT                   :rem 183
478  TA=2                      :rem 104
480  FORFA=1TOPO              :rem 99
482  X1=A(PT,1,TA):Y1=A(PT,1,TA+1) :rem 130
    :rem 118
484  X2=A(PT,2,TA):Y2=A(PT,2,TA+1) :rem 67
    :rem 158
486  GOSUB500                 :rem 2
488  TA=TA+2                  :rem 117
490  NEXTFA                   :rem 106
492  NEXTPT                    :rem 86
498  GOTO800                  :rem 225
499  REM CONFRONTO CON LA SECONDA VISUALIZ :rem 116
    AZIONE :rem 172
500  X=0:Y=0:SI=1             :rem 114
510  IFX1>X2ANDY1>Y2THENS=-1 :rem 240
520  IF(X2-X1)<(Y2-Y1)ANDY1<Y2THEN525 :rem 7
    :rem 108
522  GOTO530                  :rem 252
525  IF(Y2-Y1)>(X1-X2)THEN640 :rem 107
527  SI=-1:GOTO540            :rem 113
530  IF(X2-X1)<(Y1-Y2)ANDY1>Y2THEN620 :rem 6
    :rem 124
540  IFY1=Y2ANDX1=X2THENRETURN :rem 129
550  IFY1=Y2THEN3500          :rem 126
    :rem 85
599  REM CICLI DELLE VISUALIZZAZIONI :rem 144
    :rem 216
600  FORYY=Y1TOY2STEPSI*((Y2-Y1)/(X2-X1)): :rem 193
    Y=INT(YY):X=X1:GOSUB3000:GOSUB3200:NE :rem 24
    XTTY :rem 60
610  GOTO660                  :rem 128
620  IF(Y1-Y2)>(X1-X2)THENS=-1:GOTO640 :rem 91
    :rem 29
630  GOTO540                  :rem 73
640  IFX1=X2THEN3600           :rem 73
650  FORXX=X1TOX2STEPSI*((X2-X1)/(Y2-Y1)): :rem 24
    X=INT(XX):Y=Y1:GOSUB3000:GOSUB3300:NE :rem 60
    TXXX :rem 128
660  RETURN                   :rem 91
799  REM VA AL MENU VISUALIZZAZIONI :rem 29
    :rem 29
800  AS=""                    :rem 126
810  GETAS:IFAS=""THEN810      :rem 85
820  IFAS<>CHR$(77)THEN810     :rem 144
830  FORI=4096TO7615:POKEI,0:NEXT:rem 216
840  POKE36865,25:POKE36867,46:POKE36869,2 :rem 193
    40:POKE36879,27 :rem 24
850  PRINTCHR$(147)           :rem 60
860  GOTO50                    :rem 128
898  END                       :rem 91
999  REM COPIA UNA PARTE :rem 29
1000 PRINT"[CLR]":PRINT"PARTE";PT-1;"/TRA :rem 29
    SLAZIONE:" :rem 29
1005 PRINT"DESTRA="+X SINISTRA="-X" :rem 73
    :rem 73

```



```

1010 PRINT"BASSO +=Y ALTO{ 4 SPAZI}=-Y":P
      RINT                                     :rem 134
1015 INPUT"TRASLAZIONE X ";TX                 :rem 38
1020 INPUT"TRASLAZIONE Y ";TY                 :rem 36
1025 PRINT"LA PARTE E' TRIDIMEN- SIONALE?
      (S/N)":GOSUB3700                         :rem 134
1030 IFQS="S"THEN1100                         :rem 141
1035 A(PT,0,0)=A(PT-1,0,0):A(PT,0,1)=A(PT
      -1,0,1)                                   :rem 243
1040 TA=2                                       :rem 201
1045 FORT=2TOA(PT,0,0)+1                       :rem 97
1050 A(PT,0,TA)=A(PT-1,0,TA)+TX               :rem 222
1055 A(PT,0,TA+1)=A(PT-1,0,TA+1)+TY           :rem 156
                                           :rem 139
1060 TA=TA+2                                   :rem 95
1065 NEXTT                                     :rem 51
1068 PRINT"{CLR}"                             :rem 168
1070 RETURN                                    :rem 168
1100 A(PT,0,0)=A(PT-1,0,0):A(PT,0,1)=A(PT
      -1,0,1)                                   :rem 236
1105 FORTT=1TO2                               :rem 153
1110 TA=2                                       :rem 199
1115 FORT=2TOA(PT,0,0)+1                       :rem 95
1120 A(PT,TT,TA)=A(PT-1,TT,TA)+TX             :rem 204
                                           :rem 138
1125 A(PT,TT,TA+1)=A(PT-1,TT,TA+1)+TY         :rem 137
                                           :rem 93
1130 TA=TA+2                                   :rem 173
1135 NEXTT                                     :rem 44
1140 NEXTTT                                    :rem 171
1142 PRINT"{CLR}"                             :rem 10
1145 RETURN                                    :rem 190
1399 REM TRASLAZIONI
1400 PRINT"{CLR}":PRINT"TRASLAZIONI:"
                                           :rem 77
1405 PRINT"DESTRA+=X SINISTRA=-X"
                                           :rem 138
1410 PRINT"BASSO +=Y ALTO{ 4 SPAZI}=-Y":P
      RINT                                     :rem 42
1415 INPUT"TRASLAZIONE X ";TX                 :rem 40
1420 INPUT"TRASLAZIONE Y ";TY                 :rem 230
1430 P1=A(0,0,0):PA=P1+2                     :rem 71
1435 FORPT=2TOPT+1                             :rem 147
1440 PO=A(PT,0,0)                             :rem 210
1445 TA=2                                       :rem 18
1450 FORT=2TOPO+1                             :rem 137
1455 A(PT,0,TA)=A(PT,0,TA)+TX                 :rem 62
1460 A(PT,0,TA+1)=A(PT,0,TA+1)+TY             :rem 148
                                           :rem 95
1465 TA=TA+2                                   :rem 180
1470 NEXTT                                     :rem 7
1475 NEXTPT                                    :rem 106
1480 P2=A(1,0,0)                             :rem 152
1485 FORPT=PTOP2+PA-1                         :rem 165
1490 PO=A(PT,0,0)                             :rem 202
1495 FORTT=1TO2                               :rem 19
1500 TA=2                                       :rem 113
1505 FORT=2TOPO+1                             :rem 47
1510 A(PT,TT,TA)=A(PT,TT,TA)+TX               :rem 140
1515 A(PT,TT,TA+1)=A(PT,TT,TA+1)+TY
                                           :rem 140
1520 TA=TA+2

```

```

1525 NEXTT                                     :rem 96
1530 NEXTTT                                    :rem 176
1535 NEXTPT                                    :rem 177
1540 GOTO50                                    :rem 104
1599 REM LETTURA                             :rem 221
1600 PRINT"{CLR}":PRINT"NASTRO O DISCO":I
      NPUTOUS                                   :rem 96
1601 PRINT"NOME DISTINTIVO DEL{ 3 SPAZI}
      DISEGNO";INPUTFS                         :rem 34
1602 IFLEFTS(OUS,1)="N"THENOU=1:GOTO1605
                                           :rem 108
1603 OU=8                                       :rem 227
1605 OPEN1,OU,0,FS                             :rem 241
1610 INPUT#1,A(0,0,0),A(1,0,0)               :rem 249
1615 FORPT=2TOA(0,0,0)+1                     :rem 64
1620 TA=2                                       :rem 205
1625 INPUT#1,A(PT,0,0),A(PT,0,1)             :rem 231
1630 FORT=1TOA(PT,0,0)+1                     :rem 96
1635 INPUT#1,A(PT,0,TA),A(PT,0,TA+1)
                                           :rem 13
1640 TA=TA+2                                   :rem 143
1645 NEXTT                                     :rem 99
1650 NEXTPT                                    :rem 175
1655 FORPT=A(0,0,0)+2TOA(1,0,0)+A(0,0,0)+
      1                                         :rem 143
1657 INPUT#1,A(PT,0,0),A(PT,0,1)             :rem 236
1660 FORTT=1TO2                               :rem 159
1665 TA=2                                       :rem 214
1675 FORT=2TOA(PT,0,0)+1                     :rem 106
1680 INPUT#1,A(PT,TT,TA),A(PT,TT,TA+1)
                                           :rem 253
1685 TA=TA+2                                   :rem 152
1690 NEXTT                                     :rem 99
1700 NEXTTT                                    :rem 175
1710 NEXTPT                                    :rem 172
1715 CLOSE1                                    :rem 117
1720 GOTO50                                    :rem 104
1799 REM FILE
1800 PRINT"{CLR}":PRINT"NASTRO O DISCO":I
      NPUTINS                                   :rem 85
1801 PRINT"NOME DISTINTIVO DEL{ 3 SPAZI}
      DISEGNO";INPUTFS                         :rem 36
1803 IFLEFTS(INS,1)="N"THENIN=1:GOTO1805
                                           :rem 87
1804 IN=8                                       :rem 217
1805 OPEN1,IN,1,FS                             :rem 231
1810 RS=CHRS(13)                              :rem 51
1815 PRINT#1,A(0,0,0);RS;A(1,0,0)
                                           :rem 189
1820 FORPT=2TOA(0,0,0)+1                     :rem 62
1825 TA=0                                       :rem 210
1830 FORT=0TOA(PT,0,0)+1                     :rem 97
1835 PRINT#1,A(PT,0,TA);RS;A(PT,0,TA+1)
                                           :rem 204
1840 TA=TA+2                                   :rem 145
1845 NEXTT                                     :rem 101
1850 NEXTPT                                    :rem 177
1855 FORPT=A(0,0,0)+2TOA(1,0,0)+A(0,0,0)+
      1                                         :rem 145
1857 PRINT#1,A(PT,0,0);RS;A(PT,0,1)
                                           :rem 171
1860 FORTT=1TO2                               :rem 161
1865 TA=2                                       :rem 216
1870 FORT=2TOA(PT,0,0)+1                     :rem 103
1880 PRINT#1,A(PT,TT,TA);RS;A(PT,TT,TA+1)
                                           :rem 188
1885 TA=TA+2                                   :rem 154
1890 NEXTT                                     :rem 101
1895 NEXTTT                                    :rem 190
1900 NEXTPT                                    :rem 173
1910 CLOSE1                                    :rem 114
1915 GOTO50                                    :rem 110

```

2999 REM SUBROUTINE DI TRACCIAMENTO

```

:rem 216
3000 IFX>176ORX<0THENRETURN :rem 250
3010 IFY>160ORY<0THENRETURN :rem 246
3020 CH=INT(Y/16)*22+INT(X/8) :rem 108
3030 RO=(Y/16-INT(Y/16))*16:BY=4096+16*CH
+RO :rem 197
3040 BI=7-(X-INT(X/8)*8):POKEBY,PEEK(BY)O
R(2*BI):RETURN :rem 142
3199 REM SUBROUTINE CICLO Y :rem 141
3200 IFX1<X2THENX1=X1+1:RETURN :rem 151
3210 X1=X1-1:RETURN :rem 141
3299 REM SUBROUTINE CICLO X :rem 141
3300 IFY1<Y2THENY1=Y1+1:RETURN :rem 156
3310 Y1=Y1-1:RETURN :rem 144
3499 REM Y1=Y2 :rem 15
3500 IFX1>X2THENS1=-1 :rem 14
3510 FORXX=X1TOX2STEPX1:Y=INT(XX):Y=Y1:
GOSUB3000:NEXTXX :rem 65
3520 RETURN :rem 170
3599 REM X1=X2 :rem 14
3600 IFY1>Y2THENS1=-1 :rem 17
3610 FORYY=Y1TOY2STEPY1:Y=INT(YY):X=X1:
GOSUB3000:NEXTYY :rem 73
3620 RETURN :rem 171
3699 REM QS :rem 52
3700 QS="" :rem 192
3702 GETQS:IFQS=""THEN3702 :rem 219
3710 RETURN :rem 171
3799 REM FINE :rem 226
3800 PRINT"{CLR}{BLU}":POKE36879,27:END
:rem 110

```

Nota per l'uso della cassetta allegata:

Tutte le cassette allegate a SUPER-VIC sono prodotte con apparecchiature professionali usando nastri di ottima qualità.

Non presentano quindi alcun tipo di imperfezione che possa inibire il corretto caricamento dei programmi.

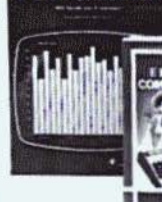
Se si dovessero riscontrare inconvenienti quali messaggi di "LOAD ERROR" oppure apparizione di caratteri strani sul video, ciò significa che le testatine del registratore usato sono sporche oppure con un allineamento non standard.

Pulitele con un po' di alcool e riallineatele operando sulla vite dell'azimut.



Libri firmati JACKSON

VOI E L'INFORMATICA



Alan Miller
PROGRAMMI SCIENTIFICI IN PASCAL

Un'opera base per chi desidera costruirsi una "libreria" di programmi in grado di risolvere i più frequenti problemi scientifici e ingegneristici. 372 pagine L. 25.000 Codice 554P

Franco Filippazzi
Giulio Occhini
VOI E L'INFORMATICA

L'opera che il manager moderno non può ignorare. In 100 tavole: gli strumenti dell'informatica, l'informatica e l'Azienda, realtà e prospettive tecnologiche... 116 pagine L. 15.000 Codice 526A

Roland Dubois
CAPIRE I MICROPROCESSORI
Un fantastico viaggio alla scoperta del "cervello" elettronico: la funzione del microprocessore, delle memorie ROM e RAM, delle interfacce... 126 pagine L. 10.000 Codice 342A

TRASMISSIONE



Giuseppe Saccardi
TRASMISSIONE DATI

Dispositivi standard e protocolli. Il calcolatore e le sue infinite applicazioni nel campo delle comunicazioni applicate a tutti i settori in cui si articola la società moderna. Un libro che traduce in tecnologia la profezia orwelliana di "1984". 308 pagine L. 23.000 Codice 528P

F. Franceschini
F. Paterlini
Voi e il vostro Commodore 64
Uno strumento fondamentale per la comprensione e programmazione del Commodore 64. Con consigli, programmi testati, glossario e utili accenni di BASIC. 256 pagine L. 22.000 Codice 347 B

La Biblioteca che fa testo

In busta chiusa, e senza impegno, inviate questo coupon a:

Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

☐ Desidero ricevere gratuitamente il Catalogo Generale della Biblioteca Jackson e informazioni sulle 10 Riviste specialistiche da voi pubblicate.

(allego L. 1.000 in francobolli per contributo spese di spedizione)

☐ Desidero ricevere contrassegno il/i volume/i

(pagherò al ricevimento L. più L. 2.000 per contributo spese di spedizione)

Nome Cognome

Via

CAP Città



Trenchfire per C64

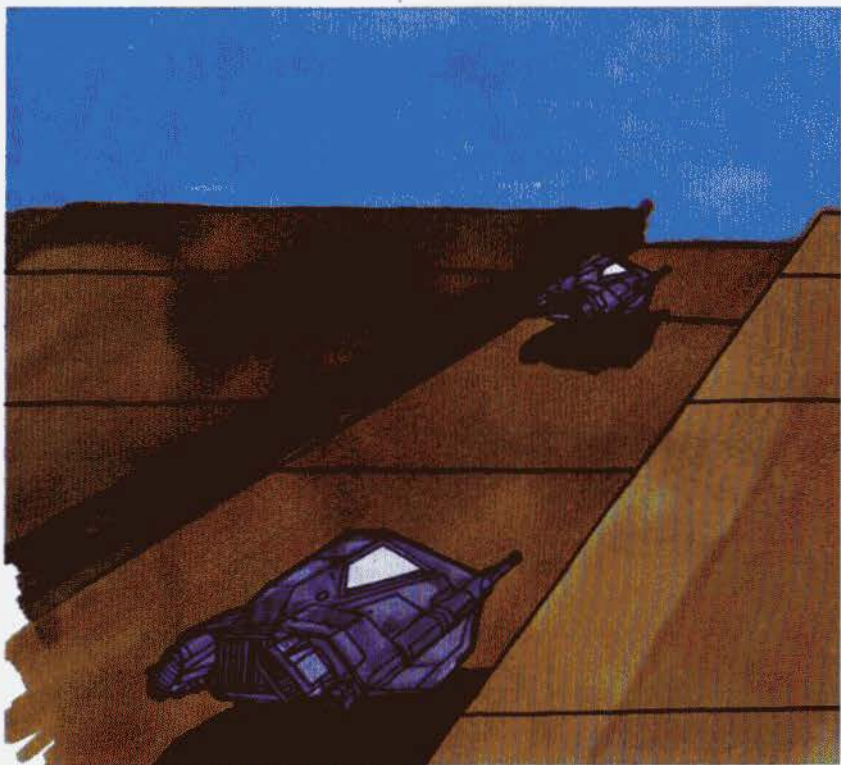
di D. Gibson
trad. ed adatt.
di M. Critulb Grizzi
e F. Sarcina

"TRENCHFIRE" è un gioco spaziale ad alta velocità che sfrutta le caratteristiche del linguaggio macchina, la potenza della grafica con SPRITE, ed uno speciale sistema di simulazione del movimento.

Con l'inizio del gioco, vi ritroverete su un lontano pianeta, volando con la vostra astronave ad altissima velocità in un canale formato da chissà quali primordiali sconvolgimenti del suolo. Lo scopo della missione sul pianeta Krypos è quello di liberare il vostro comandante catturato dai perfidi alieni; ma innanzi vi trovate faccia a faccia con i mortali caccia spaziali dei kamikaze kryponiani. La battaglia sembra non finire mai: abbattete od evitate i nemici ma vi trovate a doverne affrontare un'altra ondata: solo una concentrazione totale ed una grande rapidità di riflessi vi porteranno al successo in "TRENCHFIRE".

Scritto in BASIC con più routines in linguaggio macchina, il programma semplicemente richiede un RUN dopo che lo abbiate battuto e opportunamente salvato. Con il joystick nella porta 1 potete colpire e distruggere le navi nemiche prima che si avvicinano troppo, oppure evitarle muovendovi a destra a sinistra. Si inizia il combattimento con tre navi a disposizione, e per ogni 1000 punti ottenuti vi sarà dato un bonus di una nave (fino ad un massimo di sette). La simulazione del movimento in "TRENCHFIRE" è ottenuta cambiando ripetutamente i colori di uno sfondo precedentemente tracciato: il procedimento usa caratteri predefiniti in modo multicolore, ed una breve routine in linguaggio macchina per cambiare velocemente i registri del colore di sfondo. Un'ulteriore routine in LM si occupa della lettura del joystick al fine di ottenere una risposta immediata nel movimento.

Nota: può capitare, la prima volta che si esegue il programma, che la shape dei missili si trasformi in quella dell'e-



sione. Questo a causa di certi byte random che si trovano nella memoria del vostro C64. In ogni caso, se si do-

vesse presentare, questo inconveniente scompare la seconda volta che si esegue il programma.

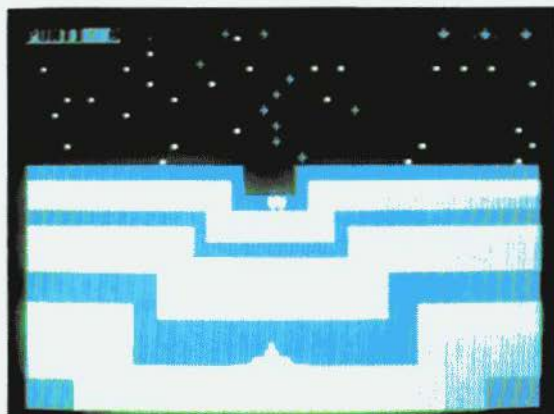
Trenchfire per C64

```
5 PRINT "{CLR}":POKE214,10:PRINT:PRINTTAB(
  13){RVS}TRENCH FIRE":FORJ=1TO2000:NEXT
      :rem 55
10 PRINT "{CLR}":POKE53281,11:POKE53282,12
  :POKE53283,13:Z=53270:POKEZ,PEEK(Z)OR1
  6      :rem 82
```

```

20 POKE52,56:POKE56,56:CLR:POKE56334,PEEK
(56334)AND254:POKE1,PEEK(1)AND251
:rem 112
30 IFPEEK(14336)<>60THENFORI=0TO511:POKEI
+14336,PEEK(I+53248):NEXT :rem 194
35 POKE1,PEEK(1)OR4:POKE53280,0:V=53248
:rem 10
40 POKE56334,PEEK(56334)OR1:POKE53272,(PE
EK(53272)AND240)+14:POKE54296,0
:rem 59
50 FORJ=0TO7:POKE14336+27*8+J,170:POKE143
36+29*8+J,85:READQ:POKE14848+J,Q
:rem 241
55 POKE14336+28*8+J,255:POKE14336+31*8+J,
255-PEEK(14336+46*8+J):NEXT :rem 239
57 DATA255,243,243,192,243,243,255,255
:rem 142
60 FORJ=0TO15:READQ:POKE14856+J,Q:NEXT:DA
TA254,254,252,240,232,156,235,247
:rem 231
61 DATA127,127,63,15,23,57,215,239
:rem 191
85 D=55296-1024:FORJ=1024TO1503:IFRND(1)>
.9THENPOKEJ,31:POKEJ+D,8:J=J+1
:rem 193
87 POKEJ,28:POKEJ+D,8:NEXT:QS="PUNTI"
:rem 61
88 FORJ=1TO5:POKE1023+J,ASC(MIDS(QS,J,1))
-64:POKE1023+J+D,0:NEXT:POKE1029,58
:rem 195
89 FORJ=1029TO1035:POKEJ+D,0:NEXT:NS=3:GO
SUB5000:SC=0:GOSUB5010 :rem 149
90 FORJ=1TO10:READQ:POKEQ-47,64:NEXT
:rem 20
95 DATA1087,1090,1165,1212,1251,1290,1331
,1297,1371,1413 :rem 6
110 FORJ=1TO16 :rem 59
113 G=0 :rem 73
115 IFJ>5THENIFJ/2=INT(J/2)THENF=F+40:G=4
0 :rem 25
120 READC :rem 242
130 T=1+J :rem 203
140 FORX=0TO18-T:M=1344+J*40+X:N=1344+J*4
0+39-X :rem 148
150 Q=M:GOSUB500:Q=N:GOSUB500 :rem 37
160 NEXTX :rem 46
170 FORI=1364+J*40TO1444+J*40+FSTEP40
:rem 90
180 Q=I-T-1:GOSUB500:Q=I+T:GOSUB500
:rem 125
190 NEXT:I=I-40 :rem 115
200 FORH=I-T-1TOI+T :rem 145
210 Q=H:GOSUB500:Q=H-G:GOSUB500:NEXT:NEXT
:GOTO700 :rem 135
300 DATA32,27,29,32,27,29,29,32,32,27,27,
29,29,29,32,32 :rem 197
500 IFQ<2024THENPOKEQ,C:POKEQ+D,8
:rem 36
501 RETURN :rem 118
700 IFPEEK(832)=33THEN800 :rem 163
705 FORJ=0TO2:FORI=0TO62:READQ:POKE832+J*
64+I,Q:NEXT:NEXT :rem 138
710 DATA33,0,0,64,128,0,140,64,0,158,64,0
,243,192,0,158,64,0,140,64,0:rem 80
720 DATA64,128,0,33,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0 :rem 238
730 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0 :rem 20
735 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0 :rem 25

```



```

740 DATA0,8,0,0,8,0,0,28,0,0,28,0,1,255,1
92,3,255,224,127,127,127 :rem 135
750 DATA30,62,60,3,99,96,1,193,192,0,0,0,
0,0,0,0,0,0,0,0 :rem 21
770 DATA0,128,0,32,32,0,0,2,0,10,128,2,40
,2,0,152,130,128,96,128,130 :rem 8
780 DATA128,150,136,0,1,215,64,215,64,128
,125,10,0,20,130,0,0,128,96,105,0
:rem 67
785 DATA0,128,128,130,128,128,41,0,64,6,0
,64,10,2,128,128,8,2,0,8,0 :rem 248
790 FORJ=15232TO15296:POKEJ,0:NEXT:FORJ=1
5253TO15273:READQ:POKEJ,Q:NEXT
:rem 218
792 DATA0,112,0,0,136,0,1,36,0,1,116,0,1,
36,0,0,136,0,0,112,0 :rem 149
800 V=53248:POKEV+21,0:POKE2040,14:POKEV+
39,1:POKEV+28,0:POKEV+46,0 :rem 173
805 POKE2047,238:POKEV+16,0:POKEV+23,0:PO
KEV+29,0:FORJ=1TO6:POKE2040+J,13
:rem 81
810 POKEV+39+J,INT(RND(1)*3)+5:NEXT
:rem 46
815 FORJ=54272TO54299:POKEJ,0:NEXT
:rem 72
900 IFV<>53248THENSTOP :rem 115
910 X=160:POKEV,X:POKEV+1,200:POKEV+29,1:
POKEV+23,1:POKEV+21,1:LV=1:W=4:E=8
:rem 42
920 POKE2047,238:IFPEEK(49172)<>206THENGOS
SUB2000 :rem 41
925 POKE49152,6:POKE49153,6:POKEV+46,0
:rem 183
930 SYS53000 :rem 147
933 S=54272:POKEV+24,15:POKEV+5,1:POKEV+6
,128:POKEV+4,129:POKEV+1,12 :rem 249
980 A=0 :rem 79
990 IFA>11+LV*3THEN6000 :rem 49
995 IF(PEEK(56321)AND16)=0THENGOSUB6200
:rem 28
1000 FORJ=1TO6:IFRND(1)<(1-LV/20)ORX(J)<>
0THEN1050 :rem 185
1005 POKE2040+J,13:X(J)=180:Y(J)=120:A=A+
1 :rem 22
1010 M(J)=RND(1)*(5+LV/2)-2-LV/4:POKEV+3
9+J,INT(RND(1)*3+0):R(J)=W :rem 237

```

```

1020 POKEV+J*2,X(J):POKEV+1+J*2,Y(J):POKE
V+21,PEEK(V+21)OR(2+J):GOTO1080
:rem 31
1050 NEXT :rem 5
1080 FORJ=1TO6:IFX(J)=0THENNEXT:GOTO1110
:rem 14
1085 IFY(J)<160ORR(J)=ETHEN1095 :rem 50
1090 POKEV+29,PEEK(V+29)OR(2+J):POKEV+23,
PEEK(V+23)OR(2+J):R(J)=E:M%(J)=M%(J)
*2 :rem 120
1092 FORN=S+7TOS+13:POKEN,0:NEXT:rem 4
1093 POKES+24,15:POKES+12,207:POKES+13,0:
POKES+8,10:POKES+11,33 :rem 175
1095 Y(J)=Y(J)+R(J):X(J)=X(J)+M%(J)
:rem 161
1100 POKEV+J*2,X(J):POKEV+1+J*2,Y(J)
:rem 67
1104 IFY(J)<222ANDPEEK(2040+J)=13THEN1107
:rem 56
1105 X(J)=0:POKEV+21,PEEK(V+21)AND(255-2+
J) :rem 193
1106 POKEV+29,PEEK(V+29)AND(255-2+J):POKE
V+23,PEEK(V+23)AND(255-2+J):rem 214
1107 NEXT :rem 8
1110 IF(PEEK(V+30)AND1)<>1THEN990
:rem 125
1111 POKEV+21,1:POKES+11,0 :rem 217
1120 POKES+1,4:FORI=1TO10:POKES+24,15-ABS
(6-I):FORJ=0TO7:POKEV+39,J :rem 59
1130 FORH=1TO10:NEXT:NEXT:NEXT:POKES+24,1
5:POKES+1,12 :rem 87
1140 GOSUB5020:K=PEEK(V+30):GOTO1000
:rem 93
2000 J=0:READT:IFT<999THENSTOP :rem 156
2002 READQ:IFQ=0THENPOKEJ+T,Q:J=J+1:GOTO
2002 :rem 56
2004 IFQ<-99THENGOTO2000 :rem 233
2006 RETURN :rem 168
2020 DATA49172,206,0,192,173,0,192,240,3,
76,100,192 :rem 247
2030 DATA173,1,192,141,0,192,162,1,254,32
,208,189,32,208,41,255,233,253,48,9
:rem 202
2040 DATA222,32,208,222,32,208,222,32,208
:rem 255
2050 DATA232,224,4,208,230,76,100,192,-1
:rem 203
2080 DATA49252,173,1,220,41,4,208,15,173,
0,208 :rem 0
2090 DATA233,105,48,8,173,0,208,233,4,141
,0,208 :rem 44
3000 DATA173,1,220,41,8,208,15,173,0,208
:rem 203
3010 DATA233,218,16,8,173,0,208,105,4,141
,0,208,76,49,234,-1 :rem 164
3050 DATA53000,120,169,20,141,20,3,169,19
2,141,21,3,88,96,-99 :rem 232
5000 IFNS>6THENNS=6:SC=SC+375:GOSUB5010
:rem 153
5001 FORJ=1062TO1065-NS*3STEP-3:POKEJ,65:
POKEJ+1,66:POKEJ+D,0:POKEJ+1+D,0:NEX
T :rem 245
5002 RETURN :rem 167
5010 $$=STR$(SC):FORJ=1TOLEN($$):POKE1029
+J,ASC(MID$($$,J,1)):NEXT :rem 177
5013 O=INT(SC/1000):IFO>PTHEPNP=O:NS=NS+1:
GOSUB5000:GOSUB5050 :rem 224
5016 RETURN :rem 172
5020 FORI=1TO2:POKE1064-NS*3+I,28:NEXT:NS
=NS-1:IFNS>0THENRETURN :rem 87
5030 PRINT" { 10 GIU' } { 14 DES } GAME OVER":

```

```

FORQQ=1TO5000:NEXT :rem 65
5031 POKE56334,PEEK(56334)AND254:rem 18
5032 POKE788,49:POKE789,234 :rem 15
5033 POKE56334,PEEK(56334)OR1:POKE198,0:P
OKE53249,0:POKE54296,0 :rem 122
5034 FORI=1TO2000:NEXT:SYS2048 :rem 79
5050 FORN=S+14TOS+20:POKEN,0:NEXT
:rem 46
5051 POKES+24,15:POKES+19,61:POKES+15,30:
POKES+18,17:RETURN :rem 46
6000 POKEV+21,1:FORJ=5TO1STEP-1:POKE49153
,J:FORI=132-J*20TO142-J*20:POKES+1,I
:rem 112
6010 FORH=1TO40:NEXT:NEXT:NEXT:SC=SC+LV*7
5:GOSUB5010:FORJ=1TO3000:NEXT:LV=LV+
1 :rem 43
6020 FORJ=2TO6:POKE49153,J:FORI=142-J*20T
O132-J*20STEP-1:POKES+1,I:FORH=1TO40
:rem 168
6030 NEXT:NEXT:NEXT:W=W+1:E=E+2 :rem 217
6040 GOTO980 :rem 164
6200 M%=(166-PEEK(V))/7:POKEV+14,PEEK(V)+
12+M%:POKEV+15,195:I=3 :rem 68
6210 POKEV+21,PEEK(V+21)OR128:FORJ=195TO1
30+LV*2STEP-3 :rem 225
6220 POKEV+15,J:I=I+1:IFI=5THENI=0:POKEV+
14,PEEK(V+14)+M% :rem 152
6230 K=PEEK(V+30):IF(KAND128)=0THENNEXT:G
OTO6300 :rem 151
6234 FORN=S+14TOS+20:POKEN,0:NEXT
:rem 51
6235 POKES+24,15:POKES+19,14:POKES+15,2:P
OKES+18,129 :rem 26
6240 FORJ=1TO6:IF(KAND(2+J))=0THENNEXT
:rem 130
6250 POKE2040+J,15:SC=SC+25:GOSUB5010
:rem 126
6300 POKEV+21,PEEK(V+21)AND127:RETURN
:rem 233

```

**Il prossimo numero di
SUPERVIC
sarà in edicola
il 10 novembre.
Non perdetelo!**



Trident per C64

di C. O. Dickerson

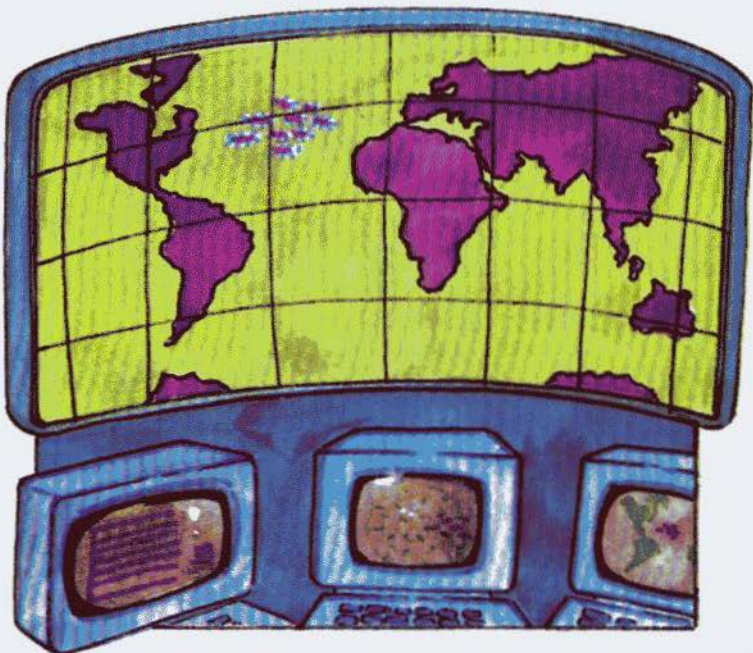
trad. e adatt.

di F. Sarcina

Siete gli ufficiali missilisti a bordo dell'U.S.S. Trident. Il più moderno e potente sottomarino nucleare mai esistito. All'improvviso, il Canale di Priorità Uno segnala un allarme rosso: il nemico ha sferrato un attacco su tutta la linea.

Voi non lo sapete, ma in realtà si tratta di un'esercitazione. Dato che il Trident è completamente computerizzato, le uniche informazioni riguardanti il mondo all'esterno del sommergibile vi giungono per mezzo del vostro video-terminale. È un semplice metodo per tenere sempre pronti gli ufficiali missilisti come voi: il computer è programmato per simulare un attacco, e quando c'è l'allarme non si può mai sapere se ciò che si osserva sullo schermo è la realtà oppure una finzione. Voi lottate non solo per il vostro Paese, ma anche per la promozione che potreste ottenere comportandovi in modo efficiente. I missili nemici giungono a ondate, in numero crescente e a velocità sempre maggiore ad ogni nuovo assalto. Dovete fronteggiare questo massiccio attacco da soli, opponendo la rapidità e la potenza del vostro computer all'assalto simulato degli ordigni nemici. I vostri missili difensivi possono volare in aria o muoversi nell'acqua a una velocità doppia di qualsiasi oggetto il nemico vi possa lanciare contro. Ma anche con armi così potenti a disposizione, vi saranno necessari riflessi prontissimi e tutta la vostra abilità per riuscire a respingere gli attacchi.

Vediamo ora come si gioca. Sullo schermo vi appaiono le segnalazioni del radar, e per neutralizzare un missile nemico in rotta di avvicinamento dovete guidare il vostro missile difensivo verso di esso per mezzo di un joystick collegato alla porta n.2. Se riuscite a raggiungere l'obiettivo, il computer vi permette di lanciare un nuovo missile difensivo. Una volta distrutta l'intera ondata, la fase successiva è più difficile in quanto, come abbiamo già accennato, i missili avversari arrivano a velocità maggiore. Se perdete (cioè uno degli ordigni nemici riesce a colpire il Trident) potete ricominciare da capo premendo il pulsante del joystick. Il



vostro punteggio è inizialmente di 2 punti per ogni missile abbattuto, e viene incrementato di uno ad ogni ondata successiva (3 punti alla seconda, 4 alla terza e così via).

Questo gioco è scritto interamente in linguaggio macchina, e deve venir memorizzato usando MLX, il programma studiato appositamente per battere senza errori i listati totalmente numerici che caratterizzano appunto il linguaggio macchina. Se non l'avete già fatto, battete e salvate su disco o nastro MLX e leggete attentamente l'articolo riguardante questo programma, in modo tale da impararne bene l'uso. A questo punto potete iniziare a memorizzare il Trident, e dato che MLX vi chiede l'indirizzo iniziale e quello finale del linguaggio macchina, sappiate che per questo gioco tali indirizzi sono rispettivamente 49152 e 51659. Quando avete finito di battere tutti quel nu-

meri (coraggio!) salvate, sempre per mezzo di MLX, il Trident su nastro o su disco. In seguito potrete caricare il gioco in memoria battendo:

LOAD "TRIDENT", 8,1 (per il disco)
oppure:

LOAD "TRIDENT", 1,1 (per il nastro)

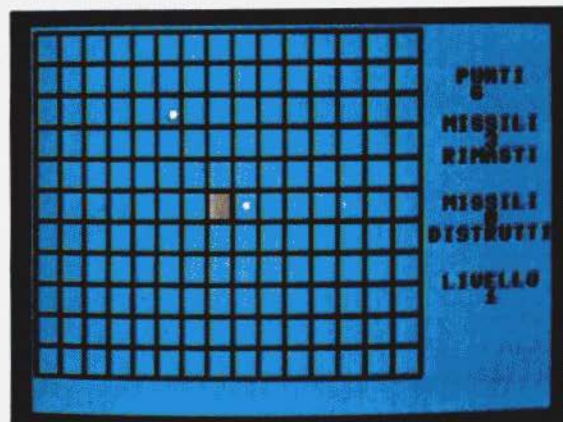
Fatto ciò, per far partire il gioco battete SYS 49152 (e non RUN!). Per concludere, sappiate che è possibile giocare a quattro diversi livelli di difficoltà, i quali determinano la velocità dei missili nemici, che è tanto maggiore quanto è più alto è il livello selezionato. La scelta avviene mediante i tasti di funzione:

F1: livello 1 - principiante
F3: livello 2 - medio
F5: livello 3 - progredito
F7: livello 4 - esperto

49152 :169,001,141,254,207,169,173
 49158 :065,141,255,207,096,169,171
 49164 :002,141,254,207,169,076,093
 49170 :141,255,207,096,169,003,121
 49176 :141,254,207,169,089,141,001
 49182 :255,207,096,169,004,141,134
 49188 :254,207,169,088,141,255,126
 49194 :207,096,165,045,133,251,171
 49200 :165,046,133,252,096,162,134
 49206 :007,230,251,208,002,230,214
 49212 :252,202,208,247,096,165,206
 49218 :251,197,047,208,006,165,172
 49224 :252,197,048,240,001,096,138
 49230 :104,104,096,160,003,177,210
 49236 :251,133,253,200,177,251,069
 49242 :133,254,096,162,000,160,127
 49248 :000,177,253,201,034,240,233
 49254 :008,157,000,206,200,232,137
 49260 :208,243,096,169,000,157,213
 49266 :000,206,096,234,160,009,051
 49272 :173,253,207,024,109,253,115
 49278 :207,136,208,249,141,253,040
 49284 :207,189,000,206,056,233,255
 49290 :048,024,109,253,207,141,152
 49296 :253,207,096,162,000,189,027
 49302 :000,206,056,233,048,141,066
 49308 :253,207,232,189,000,206,219
 49314 :201,000,240,006,032,118,247
 49320 :192,076,158,192,173,254,189
 49326 :207,201,001,208,009,173,205
 49332 :253,207,141,252,207,076,036
 49338 :220,192,201,002,208,009,250
 49344 :173,253,207,141,251,207,144
 49350 :076,220,192,201,003,208,074
 49356 :009,173,253,207,141,250,213
 49362 :207,076,220,192,173,253,051
 49368 :207,141,249,207,096,234,070
 49374 :234,234,234,234,234,234,090
 49380 :173,017,208,073,032,141,104
 49386 :017,208,173,024,208,073,169
 49392 :008,141,024,208,096,000,205
 49398 :000,000,000,000,000,173,163
 49404 :014,220,041,254,141,014,168
 49410 :220,165,001,041,251,133,045
 49416 :001,096,165,001,009,004,028
 49422 :133,001,173,014,220,009,052
 49428 :001,141,014,220,096,162,142
 49434 :000,032,251,192,189,000,178
 49440 :208,157,000,206,232,224,035
 49446 :008,208,245,032,010,193,222
 49452 :096,162,008,230,251,208,231
 49458 :002,230,252,202,208,247,167
 49464 :206,253,207,208,240,096,242
 49470 :234,234,234,169,255,141,049
 49476 :247,207,169,000,141,253,061
 49482 :207,160,008,140,254,207,026
 49488 :056,237,254,207,008,046,120
 49494 :253,207,014,255,207,042,040
 49500 :040,144,006,237,254,207,212
 49506 :076,104,193,109,254,207,017

49512 :136,208,233,176,004,109,202
 49518 :254,207,024,046,253,207,077
 49524 :096,169,000,141,254,207,215
 49530 :141,255,207,160,001,162,024
 49536 :160,238,254,207,208,003,174
 49542 :238,255,207,202,208,245,209
 49548 :136,016,240,206,253,207,174
 49554 :208,233,165,084,024,109,201
 49560 :254,207,133,084,144,002,208
 49566 :230,085,165,085,024,109,088
 49572 :255,207,133,085,096,169,085
 49578 :000,141,255,207,162,008,175
 49584 :238,255,207,202,208,250,000
 49590 :206,253,207,208,243,165,184
 49596 :084,024,109,255,207,133,232
 49602 :084,144,002,230,085,096,067
 49608 :169,000,162,128,160,096,147
 49614 :133,253,134,254,132,252,084
 49620 :169,000,133,251,160,000,157
 49626 :169,000,145,251,230,251,240
 49632 :208,002,230,252,165,251,052
 49638 :197,253,208,240,165,252,009
 49644 :197,254,208,234,032,178,059
 49650 :195,096,000,000,255,255,019
 49656 :000,000,255,255,000,032,022
 49662 :200,193,169,000,170,157,119
 49668 :000,206,157,000,207,232,038
 49674 :208,247,173,254,207,201,020
 49680 :000,208,006,032,000,192,198
 49686 :076,060,194,201,001,208,250
 49692 :006,032,011,192,076,060,149
 49698 :194,201,002,208,006,032,165
 49704 :022,192,076,060,194,201,017
 49710 :003,208,006,032,033,192,008
 49716 :076,060,194,076,102,194,242
 49722 :234,234,032,044,192,160,186
 49728 :000,177,251,205,255,207,135
 49734 :240,008,032,053,192,032,115
 49740 :065,192,208,239,200,177,133
 49746 :251,201,128,240,003,136,017
 49752 :240,238,032,081,192,032,135
 49758 :093,192,032,147,192,076,058
 49764 :012,194,032,044,192,160,222
 49770 :000,177,251,201,080,240,031
 49776 :008,032,053,192,032,065,238
 49782 :192,208,240,200,177,251,106
 49788 :201,128,240,003,136,240,048
 49794 :238,032,081,192,160,000,065
 49800 :177,253,201,034,208,003,244
 49806 :076,010,192,032,101,196,237
 49812 :234,234,201,000,208,008,009
 49818 :032,025,193,152,072,076,192
 49824 :192,194,152,072,169,208,123
 49830 :133,252,032,045,193,162,215
 49836 :000,160,000,032,251,192,039
 49842 :177,251,157,000,206,200,145
 49848 :232,224,008,208,245,032,109
 49854 :010,193,173,247,207,201,197
 49860 :000,208,046,169,000,133,240
 49866 :084,169,096,133,085,173,174

LOAD

Trident
per C64

49872 :250,207,141,255,207,032,020
 49878 :065,193,173,253,207,201,026
 49884 :000,240,022,032,117,193,056
 49890 :173,249,207,141,255,207,178
 49896 :032,065,193,173,253,207,131
 49902 :201,000,240,003,032,169,115
 49908 :193,032,178,195,173,252,243
 49914 :207,141,255,207,173,251,204
 49920 :207,141,254,207,169,000,210
 49926 :133,086,165,084,133,251,090
 49932 :165,085,133,252,238,245,106
 49938 :207,169,128,133,086,160,133
 49944 :000,162,000,138,072,189,073
 49950 :000,206,162,008,042,072,008
 49956 :144,006,032,248,195,076,225
 49962 :081,195,206,254,207,240,201
 49968 :024,070,086,165,086,201,168
 49974 :000,208,003,032,019,196,000
 49980 :173,244,207,201,000,208,069
 49986 :233,032,085,196,076,044,220
 49992 :195,070,086,173,251,207,030
 49998 :141,254,207,104,202,208,170
 50004 :205,173,246,207,201,000,092
 50010 :208,003,032,035,196,173,225
 50016 :245,207,201,008,208,003,200
 50022 :032,064,196,238,245,207,060
 50028 :165,252,133,085,165,251,135
 50034 :133,084,032,049,196,169,009
 50040 :000,141,244,207,169,128,241
 50046 :133,086,104,170,206,255,056
 50052 :207,208,148,173,252,207,047
 50058 :141,255,207,232,224,007,180
 50064 :208,137,169,000,141,244,019
 50070 :207,141,245,207,141,246,057

50076 :207,165,187,024,105,008,084
 50082 :133,084,144,002,230,188,175
 50088 :165,188,133,085,104,168,243
 50094 :200,076,136,194,162,255,173
 50100 :169,039,157,255,091,157,024
 50106 :254,092,157,253,093,157,168
 50112 :252,094,202,208,241,076,241
 50118 :119,196,173,000,221,041,180
 50124 :252,009,003,141,000,221,062
 50130 :173,024,208,041,015,009,168
 50136 :016,141,024,208,173,024,034
 50142 :208,041,240,009,004,141,097
 50148 :024,208,173,017,208,073,163
 50154 :032,141,017,208,096,000,216
 50160 :000,000,000,000,000,000,240
 50166 :000,000,070,086,165,086,141
 50172 :201,000,208,003,032,019,203
 50178 :196,017,084,145,084,206,222
 50184 :254,207,208,236,173,251,057
 50190 :207,141,254,207,096,165,060
 50196 :084,024,105,008,133,084,202
 50202 :144,002,230,085,169,128,016
 50208 :133,086,096,165,084,133,217
 50214 :187,165,085,133,188,169,197
 50220 :255,141,246,207,096,230,195
 50226 :084,208,002,230,085,165,056
 50232 :084,133,251,165,085,133,139
 50238 :252,096,160,001,162,156,121
 50244 :230,251,208,002,230,252,217
 50250 :202,208,247,136,016,242,101
 50256 :200,140,245,207,096,169,113
 50262 :128,133,086,141,244,207,001
 50268 :096,136,016,242,200,140,154
 50274 :245,207,096,072,169,000,119
 50280 :133,251,104,056,233,064,177
 50286 :176,003,024,105,064,141,111
 50292 :253,207,096,173,002,221,044
 50298 :009,003,141,002,221,173,159
 50304 :000,221,041,252,009,002,141
 50310 :141,000,221,173,024,208,133
 50316 :041,015,009,112,141,024,226
 50322 :208,173,024,208,041,240,016
 50328 :009,008,141,024,208,173,203
 50334 :017,208,009,032,141,017,070
 50340 :208,096,000,022,000,022,000
 50346 :067,003,173,062,003,074,040
 50352 :074,141,072,003,173,066,193
 50358 :003,174,067,003,024,109,050
 50364 :072,003,144,001,232,141,013
 50370 :066,003,142,067,003,162,125
 50376 :004,160,034,024,032,240,182
 50382 :255,173,067,003,174,066,176
 50388 :003,032,205,189,096,173,142
 50394 :004,208,201,138,144,003,148
 50400 :076,199,195,096,173,004,199
 50406 :208,201,138,144,248,076,221
 50412 :199,195,173,006,208,201,194
 50418 :156,176,003,076,199,195,023
 50424 :096,173,009,208,201,133,044
 50430 :144,003,076,199,195,096,199

50436 :173,011,208,201,151,176,156
 50442 :003,076,199,195,096,142,209
 50448 :060,003,140,061,003,162,189
 50454 :000,160,000,232,208,253,107
 50460 :200,204,132,003,208,247,254
 50466 :174,060,003,172,061,003,251
 50472 :096,169,147,032,210,255,181
 50478 :169,000,141,032,208,169,253
 50484 :011,141,033,208,169,154,000
 50490 :032,210,255,162,012,160,121
 50496 :016,024,032,240,255,162,025
 50502 :000,189,036,201,201,000,185
 50508 :240,007,232,032,210,255,028
 50514 :076,071,197,162,021,160,001
 50520 :007,024,032,240,255,162,040
 50526 :000,189,044,201,201,000,217
 50532 :240,007,232,032,210,255,052
 50538 :076,095,197,169,000,162,037
 50544 :000,157,000,208,232,224,165
 50550 :017,208,248,173,000,220,216
 50556 :041,016,208,237,096,000,210
 50562 :000,000,000,000,000,000,130
 50568 :000,252,019,176,195,178,188
 50574 :195,178,195,178,195,178,237
 50580 :195,178,195,178,195,178,243
 50586 :195,178,195,178,195,178,249
 50592 :195,178,195,178,195,178,255
 50598 :195,178,195,174,032,032,204
 50604 :032,032,032,032,032,032,108
 50610 :032,221,032,221,032,221,169
 50616 :032,221,032,221,032,221,175
 50622 :032,221,032,221,032,221,181
 50628 :032,221,032,221,032,221,187
 50634 :032,221,032,221,032,221,193
 50640 :032,221,032,032,032,032,077
 50646 :032,032,032,032,032,171,033
 50652 :195,219,195,219,195,219,182
 50658 :195,219,195,219,195,219,188
 50664 :195,219,195,219,195,219,194
 50670 :195,219,195,219,195,219,200
 50676 :195,219,195,219,195,179,166
 50682 :032,032,032,032,032,032,186
 50688 :032,032,032,221,032,221,058
 50694 :032,221,032,221,032,221,253
 50700 :032,221,032,221,032,221,003
 50706 :032,221,032,221,032,221,009
 50712 :032,221,032,221,032,221,015
 50718 :032,221,032,221,032,032,088
 50724 :080,085,078,084,073,032,212
 50730 :032,171,195,219,195,219,049
 50736 :195,219,195,219,195,219,010
 50742 :195,219,195,219,195,219,016
 50748 :195,219,195,219,195,219,022
 50754 :195,219,195,219,195,219,028
 50760 :195,179,032,032,032,032,062
 50766 :032,032,032,032,032,221,203
 50772 :032,221,032,221,032,221,075
 50778 :032,221,032,221,032,221,081
 50784 :032,221,032,221,032,221,087
 50790 :032,221,032,221,032,221,093

50796 :032,221,032,221,032,221,099
 50802 :032,032,032,032,032,032,050
 50808 :032,032,032,171,195,219,033
 50814 :195,219,195,219,195,219,088
 50820 :195,219,195,219,195,219,094
 50826 :195,219,195,219,195,219,100
 50832 :195,219,195,219,195,219,106
 50838 :195,219,195,179,032,077,023
 50844 :073,083,083,073,076,073,105
 50850 :032,221,032,221,032,221,153
 50856 :032,221,032,221,032,221,159
 50862 :032,221,032,221,032,221,165
 50868 :032,221,032,221,032,221,171
 50874 :032,221,032,221,032,221,177
 50880 :032,221,032,032,032,032,061
 50886 :032,032,032,032,032,171,017
 50892 :195,219,195,219,195,219,166
 50898 :195,219,195,219,195,219,172
 50904 :195,219,195,219,195,219,178
 50910 :195,219,195,219,195,219,184
 50916 :195,219,195,219,195,179,150
 50922 :032,082,073,077,065,083,134
 50928 :084,073,032,221,032,221,135
 50934 :032,221,032,221,032,221,237
 50940 :032,221,032,221,032,221,243
 50946 :032,221,032,221,032,221,249
 50952 :032,221,032,221,032,221,255
 50958 :032,221,032,221,032,032,072
 50964 :032,032,032,032,032,032,212
 50970 :032,171,195,219,195,219,033
 50976 :195,219,195,219,195,219,250
 50982 :195,219,195,219,195,219,000
 50988 :195,219,195,219,195,219,006
 50994 :195,219,195,219,195,219,012
 51000 :195,179,032,032,032,032,046
 51006 :032,032,032,032,032,221,187
 51012 :032,221,032,221,032,221,059
 51018 :032,221,032,221,032,221,065
 51024 :032,221,032,221,032,221,071
 51030 :032,221,032,221,032,221,077
 51036 :032,221,032,221,032,221,083
 51042 :032,077,073,083,083,073,007
 51048 :076,073,032,171,195,219,102
 51054 :195,219,195,219,195,219,072
 51060 :195,219,195,219,195,219,078
 51066 :195,219,195,219,195,219,084
 51072 :195,219,195,219,195,219,090
 51078 :195,219,195,179,032,032,218
 51084 :032,032,032,032,032,032,076
 51090 :032,221,032,221,032,221,137
 51096 :032,221,032,221,032,221,143
 51102 :032,221,032,221,032,221,149
 51108 :032,221,032,221,032,221,155
 51114 :032,221,032,221,032,221,161
 51120 :032,221,068,073,083,084,225
 51126 :082,085,084,084,073,171,249
 51132 :195,219,195,219,195,219,150
 51138 :195,219,195,219,195,219,156
 51144 :195,219,195,219,195,219,162
 51150 :195,219,195,219,195,219,168



```

51156 :195,219,195,219,195,179,134
51162 :032,032,032,032,032,032,154
51168 :032,032,032,221,032,221,026
51174 :032,221,032,221,032,221,221
51180 :032,221,032,221,032,221,227
51186 :032,221,032,221,032,221,233
51192 :032,221,032,221,032,221,239
51198 :032,221,032,221,032,032,056
51204 :032,032,032,032,032,032,196
51210 :032,171,195,219,195,219,017
51216 :195,219,195,219,195,219,234
51222 :195,219,195,219,195,219,240
51228 :195,219,195,219,195,219,246
51234 :195,219,195,219,195,219,252
51240 :195,179,032,076,073,086,169
51246 :069,076,076,079,032,221,087
51252 :032,221,032,221,032,221,043
51258 :032,221,032,221,032,221,049
51264 :032,221,032,221,032,221,055
51270 :032,221,032,221,032,221,061
51276 :032,221,032,221,032,221,067
51282 :032,032,032,032,032,032,018
51288 :032,032,032,171,195,219,001
51294 :195,219,195,219,195,219,056
51300 :195,219,195,219,195,219,062
51306 :195,219,195,219,195,219,068
51312 :195,219,195,219,195,219,074
51318 :195,219,195,179,032,032,202
51324 :032,032,032,032,032,032,060
51330 :032,221,032,221,032,221,121
51336 :032,221,032,221,032,221,127
51342 :032,221,032,221,032,221,133
51348 :032,221,032,221,032,221,139
51354 :032,221,032,221,032,221,145
51360 :032,221,032,032,032,032,029
51366 :032,032,032,032,032,171,241
51372 :195,219,195,219,195,219,134
51378 :195,219,195,219,195,219,140

```

```

51384 :195,219,195,219,195,219,146
51390 :195,219,195,219,195,219,152
51396 :195,219,195,219,195,179,118
51402 :032,032,032,032,032,032,138
51408 :032,032,032,221,032,221,010
51414 :032,221,032,221,032,221,205
51420 :032,221,032,221,032,221,211
51426 :032,221,032,221,032,221,217
51432 :032,221,032,221,032,221,223
51438 :032,221,032,221,032,032,040
51444 :032,032,032,032,032,032,180
51450 :032,173,195,177,195,177,175
51456 :195,177,195,177,195,177,092
51462 :195,177,195,177,195,177,098
51468 :195,177,195,177,195,177,104
51474 :195,177,195,177,195,177,110
51480 :195,189,032,032,032,032,024
51486 :032,032,032,032,032,032,190
51492 :084,082,073,068,069,078,234
51498 :084,000,032,032,032,080,046
51504 :085,076,083,065,078,084,007
51510 :069,032,080,069,082,032,162
51516 :071,073,079,067,065,082,241
51522 :069,032,032,032,032,032,199
51528 :000,000,224,000,000,224,008
51534 :000,000,000,000,000,000,078
51540 :000,000,000,000,000,000,084
51546 :000,000,000,000,000,000,090
51552 :000,000,000,000,000,000,096
51558 :000,000,000,000,000,000,102
51564 :000,000,000,000,000,000,108
51570 :000,000,000,000,000,000,114
51576 :000,000,000,000,000,000,120
51582 :000,000,000,000,000,000,126
51588 :000,000,000,127,254,000,001
51594 :127,254,000,127,254,000,132
51600 :127,254,000,127,254,000,138
51606 :127,254,000,127,254,000,144
51612 :127,254,000,127,254,000,150
51618 :127,254,000,127,254,000,156
51624 :127,254,000,127,254,000,162
51630 :127,254,000,000,000,000,043
51636 :000,000,000,000,000,000,180
51642 :000,000,000,000,000,000,186
51648 :000,000,000,000,000,000,192
51654 :008,013,013,013,013,032,034

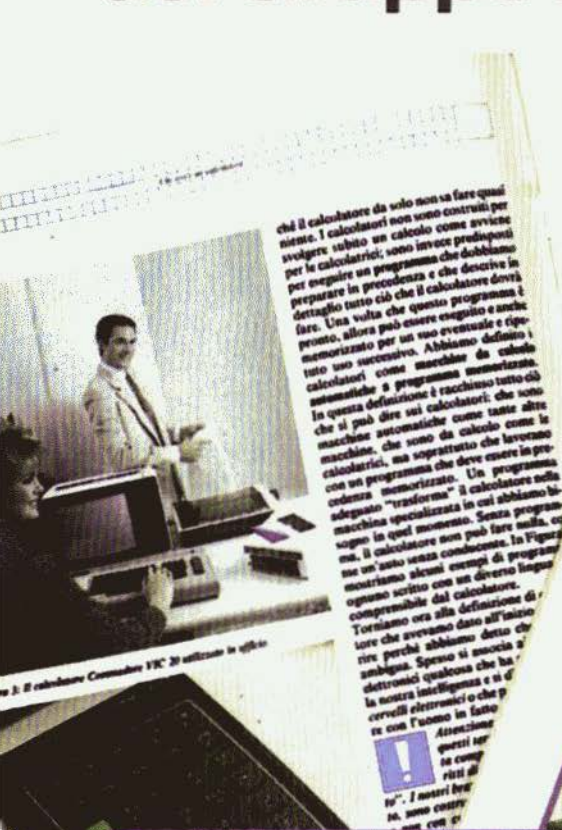
```

In seguito alle numerose telefonate prevenute, chiariamo ulteriormente la procedura di utilizzo del programma "Il signore dei draghi" di Super Vic #3. Il programma #2 non deve essere letto dal programma #1, bensì serve per creare una registrazione di dati su nastro in coda al programma #1. Poiché la cassetta allegata a Super Vic è protetta contro la registrazione occidentale, occorre duplicare il programma #1 su un'altra cassetta, quindi caricare in memoria il programma #2, rimettere nel registratore senza riavvolgerla quest'altra cassetta, e dare il RUN.

Alla fine dell'esecuzione del programma #2, su tale cassetta si troverà il programma #1 seguito dai dati necessari alla sua esecuzione, che verranno automaticamente letti dal programma #1.

ABC PERSONAL COMPUTER

il Corso di BASIC in meno di 6 mesi ideato dagli specialisti del Gruppo Editoriale Jackson



di circuiti integrati. In effetti, si potrebbe ragionare che i calcolatori sono molto stupidi, in realtà che non hanno intelligenza. La prima cosa che si deve...

conservare. La memoria centrale di un calcolatore perde infatti ogni contenuto nel momento in cui la macchina viene spenta (si dice che è una memoria volatile). È necessario quindi registrare un programma su una memoria esterna non volatile, ad esempio un disco fisso magnetico, per poterlo conservare. Per effettuare questa operazione si usa l'istruzione SAVE descritta in dettaglio nel relativo riquadro. Poiché l'uso di SAVE non è uguale in tutte le macchine, vi consigliamo di consultare il manuale del computer che state usando. Il trasferimento di un programma dalla memoria esterna a quella centrale, si effettua con l'istruzione LOAD (in alcune versioni BASIC si usa l'istruzione OLD che è equivalente a LOAD).

Sul supporto su cui si vuole salvare un programma (floppy disk, nastro, disco fisso, ecc.) deve esistere un altro programma che si chiama sistema operativo. In caso contrario, il programma non può essere eseguito. Questo sistema operativo viene fornito con il calcolatore. Quando si vuole salvare un programma, si deve prima salvare il sistema operativo. Questo sistema operativo viene fornito con il calcolatore. Quando si vuole salvare un programma, si deve prima salvare il sistema operativo. Questo sistema operativo viene fornito con il calcolatore.

Istruzioni (comandi) LOAD e OLD

LOAD [nome periferica] nome programma [nome periferica]

L'istruzione LOAD carica il contenuto di un file (programma) dalla memoria periferica (dischetto, nastro, ecc.) nella memoria centrale del calcolatore. È necessario specificare il nome del file e il nome della periferica da cui viene letto il programma.

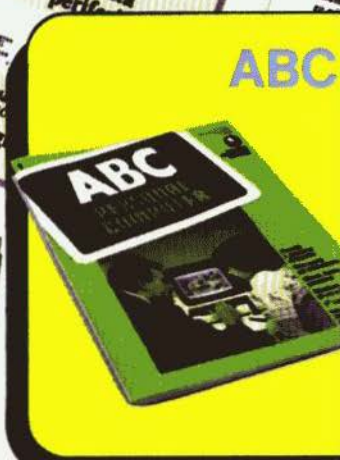
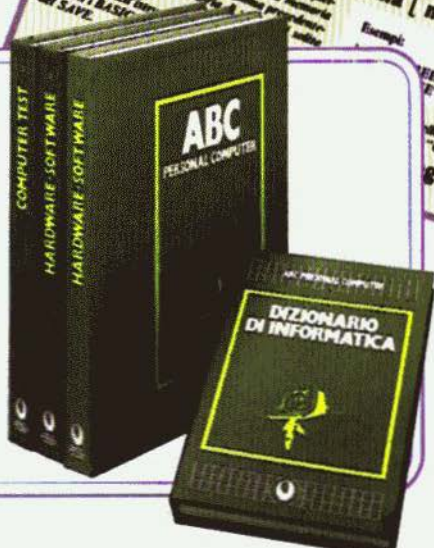
OLD [nome periferica] nome programma [nome periferica]

L'istruzione OLD è equivalente a LOAD.

ABC PERSONAL COMPUTER

Insegna tutto quel che c'è da sapere sul BASIC; mette subito in grado di programmare; fornisce una guida ragionata e competente alla scelta del Personal Computer.

ABC Personal Computer è il corso che risponde oltretutto alle esigenze di chi non ha tempo da perdere, perché:
si completa in meno di 6 mesi
si compone di 24 fascicoli settimanali



per tutti coloro che vogliono davvero "dialogare" col computer

La rapida diffusione dei computer nelle aziende, nelle scuole, nelle case ha reso urgente e indispensabile per molti imparare il linguaggio del calcolatore. Ci sono molti modi per farlo. Il più nuovo, originale, rapido e divertente si chiama...

ABC Personal Computer

l'opera creata per rispondere alle esigenze di chi, per lavoro, studio, hobby vuole acquisire un'effettiva padronanza dei piccoli elaboratori per sfruttarne le immense, entusiasmanti possibilità.

ABC Personal Computer

svela i segreti del BASIC - il linguaggio fondamentale dei personal, home e microcomputer - e fornisce la chiave per programmare da soli.

ABC Personal Computer

tratta il BASIC in modo diverso, organico, comprensibile a tutti perché nasce dalla grande esperienza e dalla capacità divulgativa del Gruppo Editoriale Jackson.

ABC Personal Computer

è un'opera preziosa, comprensibile per i ragazzi, perché comincia proprio dall'abc dell'informatica; ideale per i professionisti, perché rapida, esauriente, concreta; preziosa per gli appassionati perché ricca di programmi già testati, di suggerimenti, di notizie sul mondo dei piccoli computer.



GRUPPO EDITORIALE JACKSON

**noi l'informatica
la conosciamo davvero**

Abbonamento-risparmio

Togliendo da inviare in busta chiusa a:
Gruppo Editoriale Jackson "ABC Personal Computer"
via Rosellini, 12 - 20124 Milano

Sì, desidero sottoscrivere l'abbonamento risparmio ai 24 fascicoli di ABC Personal Computer e alle copertine dei 4 volumi dell'opera. Tutto al prezzo speciale di L. 80.000 invece di L. 96.000.

Allego alla presente

- ☐ assegno non trasferibile di L. 80.000 a voi intestato
- ☐ fotocopia di versamento di L. 80.000 sul ccp n. 11666203
- ☐ fotocopia di vaglia postale di L. 80.000 a voi intestato

I fascicoli dovranno essere inviati a:

Nome _____ Cognome _____
Via _____
Città _____ Prov. _____ C.A.P. _____
Data _____ Firma _____

Personal Computer

24 appuntamenti in edicola per arricchire la biblioteca di casa con:

- 2 volumi di Lezioni per complessive 608 pagine
- 1 volume di Computer-test
- 1 Dizionario di Informatica di oltre 208 pagine

L'opera è elegantemente rilegata in similpelle.

Centinaia di illustrazioni a colori, foto e disegni.

Formato dei volumi cm. 21x28.

Formato del Dizionario di Informatica cm. 15x21.



VIC sound

di P. Ferrari

Noise maker per C64

Quando si devono utilizzare degli effetti sonori per i propri programmi, una delle cose più noiose consiste nella ricerca per tentativi dei valori da assegnare ai vari registri del SID per ottenere una particolare sonorità.

Con il "NOISE MAKER" (lett. "Generatore di rumori"), potete variare il contenuto dei registri del SID in modo pratico e veloce allo scopo di "ritrovare" un effetto sonoro già udito (quello del vostro gioco preferito, per esempio), o di creare degli effetti nuovi ed originali.

Per meglio comprendere l'influenza dei vari parametri sul suono generato, i quattro registri che contengono ciascuno due parametri (Attack/Decay, Sustain/Release, Risonanza/Filtro, Modo-Filtro/Volume) vengono settati separatamente.

Vediamo il significato dei simboli usati:
— **W**: Forma d'onda. Viene visualizzata direttamente la forma d'onda (Δ = triangolare (17); Λ = dente di sega (33); $\square$ = quadrato (65); $\sim$ = rumore bianco (129). Essa determina la "personalità" della nota. Nel registro viene scritto il valore corrispondente alla forma prescelta.

— **A/D**: Attack e Decay. Possono assumere un valore da 0 a 15. Nel registro corrispondente (l'Attack occupa il semi-byte alto, il Decay il semi-byte basso) viene scritto il valore $16 \cdot A + D$.

— **S/R**: Sustain e Release. Come i due parametri precedenti possono variare tra 0 e 15, e nel registro corrispondente viene scritto il valore $16 \cdot S + R$.

Generalmente conviene scegliere per il Sustain un valore elevato, affinché il suono sia pienamente audibile.

— **DC1 e DC2**: Duty Cycle. Devono essere settati solo nel caso in cui si utilizzi l'onda quadrata. DC1 può variare fra 0 e 15; DC2 fra 0 e 255.

— **FON**: Accensione Filtro. L'accensione del filtro sulle tre voci viene determinata dai primi tre bit (bit 0, 1 e 2) del registro 23 del SID. Perciò, se si vuole utilizzare un filtro sulle voci 1 e 3, per esempio, bisogna porre $FON = 5$.

La parte alta del registro è usata per la Risonanza.

— **LF e HF**: Frequenza di taglio. La frequenza di taglio viene stabilita dai primi tre bit del registro 21 (LF) e dal registro 22 (HF) del SID.

— **MF**: Modo filtro. I tre tipi di filtro selezionabili (Passa-Basso, Passa-Banda e Passa-Alto) vengono attivati rispettivamente dai bit 4, 5 e 6 del registro 24. La parte basse del registro è riservata al volume.

— **RIS**: Risonanza. Occupa la parte alta del registro 23.

— **LEV**: Volume. Occupa la parte bassa del registro 24.

— **DEL**: Ritardo. Se è uguale a zero, la nota viene mantenuta finché non si suona un'altra nota. Se è maggiore di zero la nota viene suonata per un tempo proporzionale al valore prescelto (NB: Solo in questo secondo caso sarà eseguita la fase relativa al Release).

— **MOD**: Modulazione. Può agire su cinque parametri: DC1, DC2, LF, HF, e RIS. Il valore corrispondente è incrementato automaticamente mentre la nota viene suonata: e così possibile ottenere alcuni effetti in-

teressanti e divertenti.

— **TASTI CURSORE**: per passare al campo successivo premete "↓"; per tornare al campo precedente premete "↑".

Il campo corrente viene visualizzato in reverse, ad eccezione della forma d'onda (per ragioni puramente "estetiche").

— **TASTI FUNZIONE**:

— **F1/F2**: Incrementa e decrementa il campo corrente.

— **F3/F4**: Seleziona la voce.

— **F5/F6**: Trasla la tastiera di un'ottava più in alto (F5) o più in basso (F6).

— **F7**: Registra il contenuto dei campi in un file sequenziale su nastro o su disco. Il file dati così creato può essere facilmente utilizzato per i vostri programmi.

— **F8**: Carica il contenuto dei campi dal file sequenziale precedentemente registrato sul nastro o sul disco.

NB: Se premete per errore F7 e F8, per abbandonare la routine di registrazione o di lettura del file dati è sufficiente premere un altro tasto funzione qualsiasi. Se invece premete F7 al posto di F8, o viceversa basterà premere il tasto giusto per eseguire la routine desiderata.

Linee

100

200-410

420-440

450-560

570-580

600-670

700

1000-1030

1050-1130

1210-1250

1500-1550

2000-2100

3000-3070

5000-5920

6000-6900

9000-9090

9100-9400

Commento

Setta il colore del bordo, dello sfondo, dei caratteri e la ripetizione per tutti i caratteri della tastiera

Maschera dello schermo

Inizializza le variabili

Indirizzi di memoria dei Registri del SID

Azzera le forme d'onda ed il volume

Routine per la gestione della maschera

Indirizzamento diretto del cursore

Input da tastiera

Genera la nota corrispondente al tasto premuto con l'effetto di modulazione prescelto

Se necessario esegue il loop di ritardo e la fase di Release

Modulazione

Gestione dei tasti cursore e dei tasti funzione

Setta i Registri del SID

Registrazione dei dati sul nastro o sul disco

Lettura dei dati dal nastro o dal disco

Istruzioni DATA contenenti le coordinate ed il valore massimo dei campi

Istruzione DATA contenenti i valori da inserire nel byte alto e nel byte basso della frequenza

```

10 POKE53280,0:POKE53281,0:POKE646,7:POKE
650,128 :rem 237
100 PRINT"[CLR]"; :rem 48
200 PRINT"[<A>]{ 5 *}[<R>]{ 9 *}[<R>]
{ 10 *}[<R>]{ 10 *}[<S>]" :rem 87
210 PRINT"B{ 2 SPAZI}W{ 2 SPAZI}B
{ 9 SPAZI}B{ 10 SPAZI}B{ 10 SPAZI}B"
:rem 133
220 PRINT"B{ 5 SPAZI}B{ 9 SPAZI}B
{ 10 SPAZI}B{ 10 SPAZI}B" :rem 47
230 PRINT"B A/D B>{ 3 SPAZI}-{ 4 SPAZI}B
>{ 3 SPAZI}-{ 5 SPAZI}B>{ 3 SPAZI}-
{ 5 SPAZI}B" :rem 37
240 PRINT"B{ 5 SPAZI}B{ 9 SPAZI}B
{ 10 SPAZI}B{ 10 SPAZI}B" :rem 49
250 PRINT"B S/R B>{ 3 SPAZI}-{ 4 SPAZI}B
>{ 3 SPAZI}-{ 5 SPAZI}B>{ 3 SPAZI}-
{ 5 SPAZI}B" :rem 71
260 PRINT"B{ 5 SPAZI}B{ 9 SPAZI}B
{ 10 SPAZI}B{ 10 SPAZI}B" :rem 51
270 PRINT"B DC1 B>{ 8 SPAZI}B>{ 9 SPAZI}
B>{ 9 SPAZI}B" :rem 166
280 PRINT"B{ 5 SPAZI}B{ 9 SPAZI}B
{ 10 SPAZI}B{ 10 SPAZI}B" :rem 53
290 PRINT"B DC2 B>{ 8 SPAZI}B>{ 9 SPAZI}
B>{ 9 SPAZI}B" :rem 169
300 PRINT"[<Q>]{ 5 *}[<E>]{ 6 *}[<R>]
{ 2 *}[<E>]{ 8 *}[<R>]*[<E>]{ 10 *}
[<W>]" :rem 57
310 PRINT"B FON >{ 6 SPAZI}B RIS >
{ 5 SPAZI}B VOCE>{ 6 SPAZI}B"
:rem 37
320 PRINT"B LF{ 2 SPAZI}>{ 6 SPAZI}B LEV
>{ 5 SPAZI}B{ 12 SPAZI}B" :rem 99
330 PRINT"B HF{ 2 SPAZI}>{ 6 SPAZI}B DEL
>{ 5 SPAZI}B{ 12 SPAZI}B" :rem 78
340 PRINT"B MF{ 2 SPAZI}>{ 6 SPAZI}B MOD
>{ 5 SPAZI}B{ 12 SPAZI}B" :rem 95
360 PRINT"[<Q>]{ 12 *}[<E>]{ 11 *}[<E>]
{ 12 *}[<W>]:FOR I=1 TO 3 :rem 24
370 PRINT"B{ 5 SPAZI}B {RVS} {OFF} {RVS}
{OFF} B {RVS} {OFF} {RVS} {OFF} {RVS}
{OFF} B {RVS} {OFF} {RVS} {OFF} B
{RVS} {OFF} {RVS} {OFF} {RVS} {OFF} B
B{ 3 SPAZI}B":NEXT :rem 154
390 PRINT"B{ 5 SPAZI}BQBWBEBRBTBYBUBIBOB
PBEB*BtB B{ 3 SPAZI}B" :rem 64
400 PRINT"B{ 5 SPAZI}[<Z>]C[<E>]C[<E>]C
[<E>]C[<E>]C[<E>]C[<E>]C[<E>]C[<E>]C
[<E>]C[<E>]C[<E>]C[<E>]C[<E>]C[<X>]
{ 3 SPAZI}B" :rem 247
410 PRINT"[<Z>]{ 37 *}[<X>]":GOSUB800
:rem 226
420 DIMHB(95),LB(95),K(94),X(30),Y(30),M(
29),R(32) :rem 65
430 KS="Q2W3ER5T6Y7UI900P@-*+{HOME}":REM
TASTIERA :rem 255
440 FORK=1 TO 23:A=ASC(MIDS(KS,K,1)):K(A)=K
:NEXT:REM INIZIALIZZAZIONE TASTIERA
:rem 24
450 SI=54272:REM INDIRIZZO INIZIALE DEL S
ID :rem 96
460 L(1)=SI :L(2)=SI+ 7:L(3)=SI+14:REM
FREQUENZA/BYTE BASSO :rem 162
470 H(1)=SI+ 1:H(2)=SI+ 8:H(3)=SI+15:REM
FREQUENZA/BYTE ALTO :rem 173
480 B(1)=SI+ 2:B(2)=SI+ 9:B(3)=SI+16:REM
DUTY CICLE 1 :rem 50
490 C(1)=SI+ 3:C(2)=SI+10:C(3)=SI+17:REM
DUTY CICLE 2 :rem 97
500 W(1)=SI+ 4:W(2)=SI+11:W(3)=SI+18:REM
FORMA D'ONDA :rem 199

```

```

510 A(1)=-SI+ 5:A(2)=-SI+12:A(3)=-SI+19:REM
    ATTACK/DECAY :rem 207
520 S(1)=-SI+ 6:S(2)=-SI+13:S(3)=-SI+20:REM
    SUSTAIN/RELEASE :rem 10
530 LF =SI+21:REM FREQUENZA DI TAGLIO/BY
    TE BASSO :rem 136
540 HF =SI+22:REM FREQUENZA DI TAGLIO/BY
    TE ALTO :rem 62
550 FL =SI+23:REM ACCENSIONE FILTRO
    :rem 91
560 VL =SI+24:REM VOLUME :rem 157
570 FORI=1TO3:POKEW(I),0:NEXT :rem 63
580 POKEFL,0:POKEHF,0:POKEFL,0:POKEVL,0
    :rem 203
590 GOSUB9000:GOTO1000 :rem 27
600 X=X(I):Y=Y(I):REM CAMPO :rem 110
602 IFI=1ORI=8ORI=15THENS$=WS(R(I)):R=-1:
    RETURN :rem 166
603 IFI<29THEN609 :rem 36
604 S$=M$(R(I)):M=R(I):IFM=0THENRETURN
    :rem 62
605 IFM=1ORM=2THENM=J+4+M:RETURN:rem 54
606 IFM=3ORM=4THENM=20+M:RETURN :rem 244
607 M=26:RETURN :rem 169
609 S$=RIGHT$( "00"+MID$(STR$(R(I)),2),3):
    RETURN :rem 149
610 GOSUB600:S$="{OFF}"+S$:GOTO700
    :rem 29
620 IFR(I)>M(I) THENR(I)=0 :rem 192
625 IFR(I)<0THENR(I)=M(I) :rem 195
630 GOSUB600:IFR=1THENS$="{RVS}"+S$
    :rem 19
635 GOTO700 :rem 110
640 X=X(30):Y=Y(30):REM VOCE :rem 99
645 IFV>3THENV=1 :rem 232
650 IFV<1THENV=3 :rem 226
660 S$="00"+MID$(STR$(V),2):J=V*7-6
    :rem 107
670 IFR(29)=1THENM=J+6 :rem 12
700 POKE211,X:POKE214,Y:SYS58640:REM POSI
    ZIONAMENTO CURSORE :rem 35
710 PRINTS$+"{OFF}";:R=1:RETURN :rem 236
800 X=0:Y=23:S$="{RVS}UN MOMENTO, PREGO..
    .":GOTO700 :rem 213
810 X=0:Y=23:GOTO825 :rem 165
820 X=0:Y=22 :rem 147
825 S$="{ 39 SPAZI}":GOTO700 :rem 161
830 X=0:Y=22:GOTO845 :rem 168
840 X=0:Y=21 :rem 148
845 S$="":GOTO700 :rem 163
1000 GOSUB810:I=1:J=1:REM INPUT TASTIERA
    :rem 200
1010 GETA$:IFA$=""THEN1110 :rem 168
1020 A=ASC(A$) :rem 206
1030 IFA<19ORA>94THEN2000 :rem 184
1050 N=K(A) :rem 46
1060 IFTHENK=N+S*12 :rem 180
1070 POKEA(V),R(J+1)*16+R(J+2):REM ATTACK
    /DECAY :rem 197
1080 POKES(V),R(J+3)*16+R(J+4):REM SUSTAI
    N/RELEASE :rem 230
1090 POKEL(V),LB(K):POKEH(V),HB(K):REM FR
    EQUENZA :rem 187
1100 POKEW(V),16*2+R(J)+1:REM FORMA D'OND
    A :rem 165
1110 IFR(29)=0THEN1200 :rem 191
1120 KM=KM+1:IFKM>M(M) THENKM=1 :rem 76
1130 ONR(29)GOSUB1510,1520,1530,1540,1550
    :rem 141
1200 IFR(28)=0THEN1010 :rem 189
1210 FORDU=1TOR(28)*10:GOTO1230 :rem 3
1230

```

```

1220 GETAS:IFAS<>" THENPOKEW(V),16*21R(J)
      :FORDU=1TOR(28)*10:NEXT:GOTO1020
      :rem 158
1230 NEXT :rem 5
1240 POKEW(V),16*21R(J):FORDU=1TOR(28)*10
      :NEXT :rem 10
1250 GOTO1010 :rem 195
1510 POKEH(V),KM:KM=KM+9:IFKM>254THENKM=0
      :rem 28
1515 RETURN :rem 172
1520 POKEC(V),KM:RETURN :rem 191
1530 POKEF(V),KM:RETURN :rem 104
1540 POKEH(V),KM:KM=KM+9:IFKM>254THENKM=0
      :rem 196
1545 RETURN :rem 175
1550 POKEF(V),KM*16+R(22):RETURN :rem 45
2000 IFA=17ANDI<29THENGOSUB610:I=I+1:GOSU
      B620:GOTO3000:REM CURSORE IN GIU'
      :rem 17
2010 IFA=145ANDI>1THENGOSUB610:I=I-1:GOSU
      B620:GOTO3000:REM CURSORE IN SU'
      :rem 209
2020 IFA=133THENR(I)=R(I)+1:GOSUB620:GOTO
      3000:REM >F1< :rem 159
2030 IFA=137THENR(I)=R(I)-1:GOSUB620:GOTO
      3000:REM >F2< :rem 167
2040 IFA=134THENPOKEW(V),0:V=V+1:GOSUB640
      :GOTO3000:REM >F3< :rem 61
2050 IFA=138THENPOKEW(V),0:V=V-1:GOSUB640
      :GOTO3000:REM >F4< :rem 69
2060 IFA=135ANDS<6THENS=S+1:GOTO1060:REM
      >F5< :rem 193
2070 IFA=139ANDS>0THENS=S-1:GOTO1060:REM
      >F6< :rem 197
2080 IFA=136THENGOSUB5000:GOTO3000:REM >F
      7< :rem 48
2090 IFA=140THENGOSUB6000:GOTO3000:REM >F
      8< :rem 46
2100 GOTO1010 :rem 190
3000 POKEB(V),R(J+5):REM DC1 :rem 42
3010 POKEC(V),R(J+6):REM DC2 :rem 46
3020 POKEF(V),R(23):POKEH(V),R(24):REM FREQUE
      NZA FILTRO :rem 133
3030 POKEF(V),R(26)*16+R(22):REM FILTRO
      :rem 111
3040 POKEV(V),R(25)*16+R(27):REM VOLUME
      :rem 140
3070 GOTO1010 :rem 197
5000 X=0:Y=22:SS="{RVS}SAVE":GOSUB700:REM
      SAVE :rem 207
5010 X=6:Y=23:SS="{RVS}D{OFF}ISK OR {RVS}
      T{OFF}APE ":GOSUB700 :rem 135
5020 GETAS:IFAS="{RVS}THEN5020 :rem 177
5030 GOSUB5900:IFAS="{RVS}THEN5500 :rem 2
5035 IFAS="{RVS}THEN6000 :rem 195
5040 IFASC(AS)>132ANDASC(AS)<140THENRETRU
      N :rem 249
5045 IFAS<>"T"THEN5000 :rem 200
5050 GOSUB5910:GOSUB820:GOSUB810:rem 192
5060 GOSUB830:OPEN1,1,1,N$ :rem 219
5070 FORDY=1TO29:PRINT#1,R(DY):NEXT
      :rem 204
5080 CLOSE1 :rem 116
5090 GOSUB820:GOSUB810 :rem 59

```

```

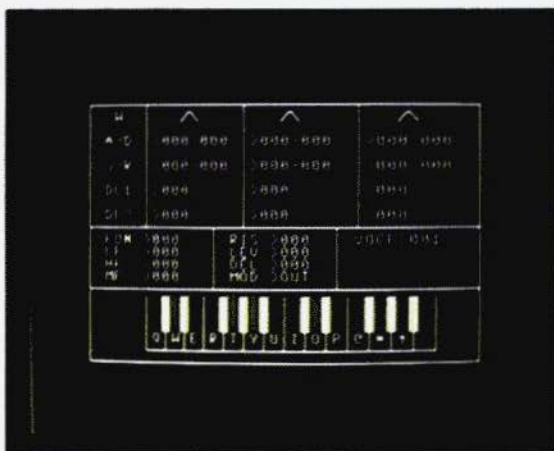
5100 RETURN :rem 166
5500 GOSUB5910 :rem 25
5510 OPEN2,8,2,"@:"+"NS"+"S,W" :rem 25
5520 FORDY=1TO29:PRINT#2,R(DY):NEXT
      :rem 205
5530 CLOSE2:GOTO5090 :rem 182
5900 PRINTAS;:FORDY=1TO300:NEXT:GOSUB820:
      GOSUB810:GOSUB830:RETURN :rem 236
5910 N$="":INPUT"NAME DEL FILE "N$
      :rem 161
5920 RETURN :rem 176
6000 X=0:Y=22:SS="{RVS}LOAD":GOSUB700:REM
      LOAD :rem 178
6010 X=6:Y=23:SS="{RVS}D{OFF}ISK OR {RVS}
      T{OFF}APE ":GOSUB700 :rem 136
6020 GETAS:IFAS="{RVS}THEN6020 :rem 179
6030 GOSUB5900:IFAS="{RVS}THEN6500 :rem 4
6035 IFAS="{RVS}THEN5000 :rem 191
6040 IFASC(AS)>132ANDASC(AS)<140THENRETRU
      N :rem 250
6045 IFAS<>"T"THEN6000 :rem 202
6050 GOSUB5910:GOSUB820:GOSUB810:rem 193
6060 GOSUB840:OPEN1,1,0,N$ :rem 220
6070 FORDY=1TO29:INPUT#1,R(DY):NEXT
      :rem 208
6080 CLOSE1 :rem 117
6090 GOSUB820:GOSUB810:GOSUB6900:rem 197
6100 RETURN :rem 167
6500 GOSUB5910 :rem 26
6510 OPEN2,8,2,"@:"+"NS"+"S,R" :rem 213
6520 FORDY=1TO29:INPUT#2,R(DY):NEXT
      :rem 209
6530 CLOSE2:GOTO6090 :rem 184
6900 IS=I:FORI=1TO29:GOSUB610:NEXT:I=IS:R
      ETURN :rem 23
9000 REM DATA :rem 199
9005 FORI=1TO7:READX(I),Y(I),M(I):X(I+7)=
      X(I)+10:X(I+14)=X(I)+21 :rem 130
9010 FORJ=1TO2:JJ=J*7:Y(I+JJ)=Y(I):M(I+JJ
      )=M(I):NEXT:NEXT :rem 6
9020 FORI=22TO29:READX(I),Y(I),M(I):NEXT
      :rem 157
9030 READX(30),Y(30) :rem 45
9040 DATA 010,001,003,008,003,015
      :rem 57
9042 DATA 012,003,015,008,005,015
      :rem 68
9044 DATA 012,005,015,008,007,255
      :rem 80
9046 DATA 008,009,015 :rem 12
9050 DATA 007,011,015,007,012,007
      :rem 68
9052 DATA 007,013,255,007,014,015
      :rem 79
9054 DATA 020,011,015,020,012,015
      :rem 61
9056 DATA 020,013,100,020,014,005
      :rem 61
9060 DATA 032,011 :rem 60
9070 FORI=0TO3:READWS(I):NEXT :rem 39
9075 DATA "NM {GIU'}{ 2 SIN}{ 2 SPAZI}",
      "N{<G>}{GIU'}{ 2 SIN}{ 2 SPAZI}",
      "[<M>]{<T>}{<G>}{GIU'}{ 2 SIN}
      { 2 SPAZI}", "UI {GIU'}{ 2 SIN}JK"
      :rem 223
9080 FORI=0TO5:READMS(I):NEXT :rem 32
9085 DATA "OUT","DC1","DC2","LF ","HF ","
      RIS" :rem 219
9090 V=1:R(30)=1:FORI=1TO30:GOSUB610:NEXT
      :rem 236

```

```

9100 FOR I=1 TO 95:READ HB(I),LB(I):NEXT:REM
      FREQUENZE (BYTE ALTO,BYTE BASSO)
      :rem 32
9110 DATA 001,022,001,039,001,057
      :rem 64
9120 DATA 001,075,001,095,001,116
      :rem 71
9130 DATA 001,138,001,161,001,186
      :rem 73
9135 DATA 001,212,001,240,002,014
      :rem 60
9140 DATA 002,045,002,078,002,113
      :rem 71
9150 DATA 002,150,002,190,002,231
      :rem 65
9160 DATA 003,020,003,066,003,116
      :rem 69
9165 DATA 003,169,003,224,004,027
      :rem 86
9170 DATA 004,090,004,156,004,226
      :rem 82
9180 DATA 005,045,005,123,005,207
      :rem 79
9190 DATA 006,039,006,133,006,232
      :rem 85
9195 DATA 007,081,007,193,008,055
      :rem 100
9200 DATA 008,180,009,056,009,196
      :rem 95
9210 DATA 010,089,010,247,011,158
      :rem 82
9220 DATA 012,078,013,010,013,208
      :rem 72
9225 DATA 014,162,015,129,016,109
      :rem 89
9230 DATA 017,103,018,112,019,137
      :rem 82
9240 DATA 020,178,021,237,023,059
      :rem 89
9250 DATA 024,157,026,020,027,160
      :rem 83
9255 DATA 029,069,031,003,032,219
      :rem 93
9260 DATA 034,207,036,225,039,018
      :rem 94

```



```

9270 DATA 041,101,043,219,046,118
      :rem 86
9280 DATA 049,058,052,039,055,065
      :rem 107
9285 DATA 058,138,062,005,065,181
      :rem 105
9290 DATA 069,157,073,193,078,036
      :rem 117
9300 DATA 082,210,087,182,092,237
      :rem 96
9310 DATA 098,115,104,078,110,130
      :rem 85
9315 DATA 117,020,124,010,131,106
      :rem 71
9320 DATA 139,059,147,130,156,072
      :rem 100
9330 DATA 165,147,175,107,185,218
      :rem 107
9340 DATA 196,231,208,156,221,004
      :rem 91
9345 DATA 234,040,248,020
      :rem 208
9400 RETURN
      :rem 173

```

**Per ricevere la cassetta contenente
i programmi di SUPERVIC
compilate e spedite il coupon qui sotto!**

SUPERVIC

CEDOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.soft - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

Vogliate inviarmi n° _____ cassetta

contenenti programmi pubblicati sul numero..... di SUPERVIC al prezzo
speciale di L. 15.000 + 2.000 per spese di spedizione ciascuna

☐ Contanti allegati ☐ Assegno allegato n° _____

☐ Ho spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul ccp. n° 19445204 intestato a J.soft - Milano

☐ Pagherò in contrassegno al postino al ricevimento della cassetta

buono d'ordine per la cassetta contenente i programmi pub-
blicati sul numero... di SUPERVIC

Nome _____

Cognome _____

Via _____

Città _____ C.A.P. _____ Prov. _____

Se richiesta lettura:

Cod. F. e P. IVA _____

Data _____

Firma _____

Per i residenti all'estero — pagamento anticipato (vaglia o versamento su ns. ccp)

VIC sound

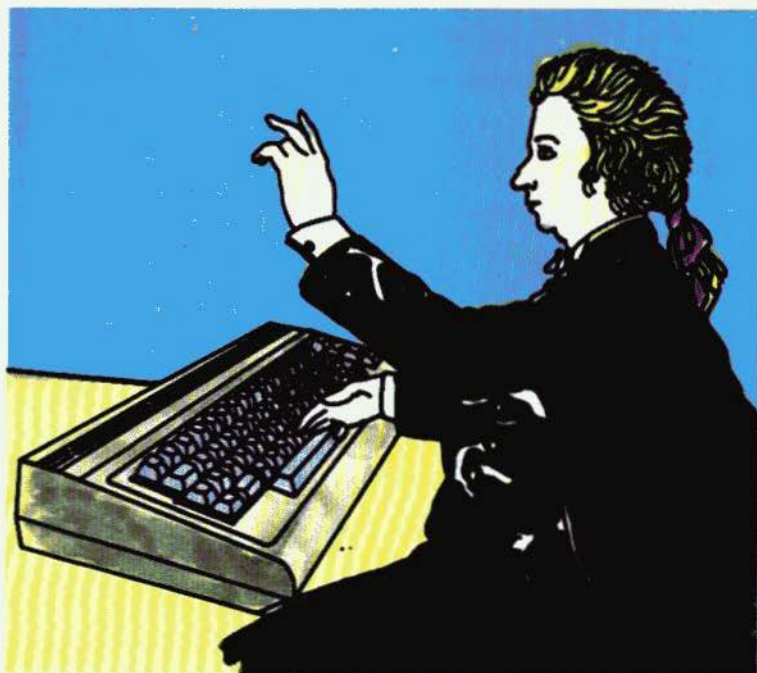
Mozart

di D. J. Eddington
trad. e adatt. di E. Comini
e M. Cristulb Grizzi

Il vostro computer sarà in grado di comporre musica con questo speciale programma, e le composizioni saranno prettamente in stile mozartiano.

Se vi siete già accinti a programmare il vostro computer per fare in modo che suoni una particolare musica, saprete che ciò implica un notevole sforzo di programmazione. Avere il proprio computer che scrive la musica è una cosa alquanto insolita.

Per ottenere ciò, abbiamo per prima cosa trovato un sistema per lavorare con delle istruzioni POKE, le quali caricano in memoria valori contenuti in linee DATA per dare l'ossatura alla musica.



Linea	Commento
10	Dimensionamento matrice delle note
30-50	Informazioni stampate sullo schermo
75	Aspetta 2 secondi prima di continuare
90	Legge nella matrice tutte le note e nella seguente istruzione DATA fino a che 14 variazioni delle 9 battute sono state lette.
91-95	Istruzioni DATA lette dalla linea 90.
100-130	Valore della velocità e numeri delle voci. La linea 130 è il volume.
160-178	Istruzioni DATA che mantengono lo stile mozartiano.
180	Legge il successivo dato
190-270	Determina il valore di Y
300-332	Queste linee garantiscono che il valore di Y sarà sempre diverso ogni volta che il programma verrà mandato in esecuzione. Questa parte è un generatore di numeri casuali.
500-510	Parte che è stampata sullo schermo quando è terminato l'ultimo brano.
1010	È qui che la musica viene creata;
1035	la linea 1035 rimanda alla battuta 160 per creare un'altra battuta di musica.
1500	Termina il brano con un lungo accordo;
1510	Poi lascia 2 secondi di intervallo tra i brani. Quindi esegue un altro brano.

Inoltre, deve essere possibile leggere questi valori in qualunque ordine, in modo che la musica sia differente ogni volta che il programma viene mandato in esecuzione. I sistemi comunemente usati di manipolazione delle stringhe non avrebbero funzionato molto bene in questo caso: occorre varietà, e questi sistemi pur molto veloci avrebbero reso il listato molto meno pulito e comprensibile.

In alternativa, si può scrivere ogni battuta come una serie di POKE nota/durata/prossima nota e così via... Ma dopo solo una dozzina di battute inserite in questo modo, sarebbe comparso sul VIC il fatidico messaggio "OUT OF MEMORY ERROR". I risultati di questo metodo sono buoni, ma probabilmente lo avreste trovato troppo lungo e ripetitivo.

Il sistema più breve e più efficiente per risolvere il problema è un particolare metodo chiamato "array referencing": per settare le battute della musica, si crea una matrice per tutte le variabili, poi si fa riferimento a questa matrice

In un loop di POKE. Specificatamente, sono richieste 14 variazioni su 9 variabili per creare la musica di questo programma. Il generatore di numeri casuali è sfruttato per rendere la musica diversa ad ogni esecuzione del programma.

Lo stile mozartiano risulta da un deli-

berato accorciamento delle note di tonalità più bassa, e dal porre le note di tonalità alta con una lunghezza variabile. Per prevenire il fatto che la musica diventi totalmente casuale, le istruzioni DATA selezionano le battute dalla loro tonalità: tonica, sottodominante,

dominante, o supertonica. È inoltre prevista una cadenza ogni quattro battute, ed un finale per ogni brano. Questa spiegazione linea per linea (della versione per VIC 20) aiuterà ad illustrare i passi di programma più significativi.

Mozart versione per VIC 20

```

10 DIMX(14,9) :rem 174
25 POKE36879,90 :rem 60
30 PRINT"[CLR]{ 2 GIU'}{DES}{WHT}BENVENUT
I! IO SONO":PRINT"{ 2 SPAZI}VICLANG
{ 2 SPAZI}AMAZIUS":PRINT"{ 7 DES}MOZAR
T" :rem 143
31 PRINT"{GIU'}{ 2 DES}COMPONGO ED ESEGUO
":PRINT"MUSICA COME QUELLA DEL":PRINT"
{SU}{ 3 DES}BAMBINO PRODIGIO":rem 5
32 PRINT"{GIU'} WOLFGANG{ 4 SPAZI}AMADEUS
":PRINT"{ 8 DES}MOZART":PRINT"{GIU'}MO
ZART VISSE DAL{ 2 SPAZI}1756":rem 147
33 PRINT"{SU}AL 1791 E{ 2 SPAZI}SCRISSE 6
26":PRINT"{SU}LAVORI IN 31 ANNI." :rem 222
34 PRINT"{GIU'}{RED}I 5{ 2 SPAZI}BRANI CH
E{ 2 SPAZI}ADESSO":PRINT"{SU}ASCOLTATE
{ 2 SPAZI}SONO{ 2 SPAZI}STATI":PRINT"
{SU}COMPOSTI DAL COMPUTER" :rem 40
80 POKE36879,15 :rem 58
90 FORT=1TO14:FORTT=1TO9:READX:X(T,TT)=X+
212:NEXTTT,T :rem 220
91 DATA3,11,11,11,11,16,16,11,7,3,11,16,1
1,16,13,16,11,16,3,11,13,11,16,13,16,1
1,16 :rem 205
92 DATA3,13,16,13,19,22,19,23,13,3,13,19,
13,19,16,19,13,13,3,19,13,13,3,19,19,1
3,3 :rem 203
93 DATA7,13,16,13,22,16,16,13,7,0,7,16,7,
13,11,16,7,13,7,19,22,13,16,13,16,11,7
:rem 50
94 DATA7,13,11,13,7,11,19,13,7,7,13,11,13,
7,11,19,13,13,3,11,13,11,16,16,16,16,
16 :rem 136
95 DATA0,16,13,7,7,7,16,7,7,3,19,16,13,13,
13,19,13,13 :rem 75
100 REM SELEZIONE DEI GENERATORI E DELLA
VELOCITA' :rem 10
120 K=36875:L=36876:P=175 :rem 93
130 POKE36878,12 :rem 98
160 REM SELEZIONE DELLE BATTUTE DAI NUMER
I DELLE LINEE SEGUENTI :rem 125
170 DATA1,3,6,2,1,4,6,2,3,4,1,5,1,4,6,7,1
,4,6,2,1,3,6,9 :rem 126
172 DATA1,1,4,5,1,4,6,2,3,4,1,5,1,4,1,5,1
,4,6,9 :rem 252
174 DATA1,4,6,2,3,6,1,5,1,4,6,7,3,4,6,2,1
,4,3,7,1,4,6,9 :rem 138
176 DATA1,4,3,7,1,6,4,5,6,3,6,2,4,6,1,5,1
,4,6,9 :rem 16
178 DATA1,4,3,7,6,3,6,2,4,6,1,5,1,3,6,7,3
,6,1,5,1,4,6,9,8 :rem 246
180 READR :rem 89
190 IFRR=1THEN300 :rem 253
200 IFRR=2THENY=12:GOTO1010 :rem 145
210 IFRR=3THEN310 :rem 249
220 IFRR=4THEN320 :rem 252
230 IFRR=5THENY=14:GOTO1010 :rem 153

```

```

240 IFRR=6THEN330 :rem 1
250 IFRR=7THENY=13:GOTO1010 :rem 156
260 IFRR=8THEN500 :rem 4
270 IFRR=9THEN1500 :rem 55
300 Y=1:X=RND(1):IFX<.35THENY=3 :rem 122
301 IFX>.75THENY=2 :rem 74
302 GOTO1010 :rem 144
310 Y=10:IFRND(1)<.4THENY=11:GOTO1010 :rem 180
320 Y=4:X=RND(1):IFX<.35THENY=5 :rem 129
321 IFX>.75THENY=6 :rem 80
322 GOTO1010 :rem 146
330 Y=7:X=RND(1):IFX<.35THENY=8 :rem 136
331 IFX>.75THENY=9 :rem 84
332 GOTO1010 :rem 147
500 PRINT"[CLR]{GIU'}{YEL} BENE,QUESTO E'
TUTTO":PRINT"SPERO VI{ 2 SPAZI}SIA PI
ACIUTO" :rem 73
510 PRINT"DATE{ 2 SPAZI}DI NUOVO IL
{ 2 SPAZI}RUN":PRINT"{SU}ED ASCOLTARE
TE ALTRI 5":PRINT"{SU}BRANI!!":END :rem 159
900 REM LE SEGUENTI SONO LE BATTUTE MUSIC
ALI CHE :rem 130
910 REM ** VICLANG UTILIZZA PER COMPORRE
L'INTERO ACCORDO :rem 223
1010 POKEK,X(Y,1):POKEL,X(Y,2):FORT=1TOP:
NEXT:POKEK,0:POKEL,X(Y,3):FORT=1TOP:
NEXT :rem 248
1020 POKEK,X(Y,4):POKEL,X(Y,5):FORT=1TOP:
NEXT:POKEK,0:POKEL,X(Y,6):FORT=1TOP:
NEXT :rem 2
1030 POKEK,X(Y,7):POKEL,X(Y,8):FORT=1TOP:
NEXT:POKEK,0 :rem 117
1035 POKEK,X(Y,9):FORT=1TOP:NEXT:GOTO160 :rem 48
1500 POKE36876,235:POKE36875,239:POKE3687
4,235:FORT=1TO1200:NEXT :rem 6
1510 POKE36874,0:POKE36875,0:POKE36876,0:
FORT=1TO2000:NEXT:GOTO160 :rem 206

```

Mozart versione per C64

```

100 DIMH(14,9),L(14,9) :rem 97
101 FORT=54272TO54272+24:POKET,0:NEXT :rem 216
102 POKE54296,15 :rem 94
103 FORT=54272+5TO54272+24STEP7:POKET,17:
POKET+1,244:NEXT :rem 196
110 POKE53281,7:POKE53280,5 :rem 243
120 PRINT"[CLR]{ 2 GIU'}{DES}{WHT}SALVE!
SONO 64CLANG AMAZIUS' MOZART." :rem 10
130 PRINT"{GIU'}SUONO MUSICHE COME IL BAM
BINO PRODIGIO," :rem 82
135 PRINT"WOLFGANG AMADEUS MOZART POTREBB
E FARE." :rem 59
140 PRINT"{ 2 GIU'}MOZART VISSE DAL 1756
AL 1791 E SCRISSE" :rem 83

```

```

145 PRINT"PIU' DI 626 OPERE IN 31 ANNI." :rem 117
150 PRINT"[GIU']{BLK}I 5 PEZZI CHE ASCOLT :rem 147
    ERAI SONO STATI" :rem 147
155 PRINT"COMPOSTI DAL COMPUTER COME TU L :rem 139
    I ASCOLTI" :rem 139
160 FOR=1TO2000:NEXT :rem 30
170 POKE53281,5:POKE53280,7 :rem 249
180 FOR=1TO14:FOR=1TO9:READH,L:H(T,TT) :rem 105
    =H:L(T,TT)=L:NEXTT:NEXTT :rem 105
190 DATA2,143,31,165,31,165,15,210,31,16 :rem 122
    5,37,162,18,209,31,165,14,24:rem 102
200 DATA2,143,31,165,37,162,15,210,37,16 :rem 153
    2,33,135,18,209,31,165,18,209 :rem 153
210 DATA2,143,31,165,33,135,15,210,37,16 :rem 150
    2,33,135,18,209,31,165,18,209 :rem 150
220 DATA2,143,33,135,37,162,16,195,42,62 :rem 206
    ,50,60,21,31,50,60,16,195 :rem 206
230 DATA2,143,33,135,42,62,16,195,42,62, :rem 6
    37,162,21,31,33,135,16,195 :rem 6
240 DATA2,143,42,62,33,135,16,195,25,30, :rem 249
    84,125,21,31,33,135,12,143 :rem 249
250 DATA4,24,33,135,37,162,16,195,50,60, :rem 60
    37,162,18,209,33,135,12,143 :rem 60
260 DATA4,24,28,49,37,162,14,24,33,135,3 :rem 238
    1,165,18,209,28,49,16,195 :rem 238
270 DATA4,24,42,62,50,60,16,195,37,162,3 :rem 218
    3,135,18,209,31,165,14,24 :rem 218
280 DATA4,24,33,135,31,165,16,195,28,49, :rem 216
    31,165,21,31,33,135,14,24 :rem 216
290 DATA4,24,33,135,31,165,16,195,28,49, :rem 20
    31,165,21,31,33,135,16,195 :rem 20
300 DATA2,143,31,165,33,135,15,210,37,16 :rem 157
    2,37,162,18,209,37,162,18,209 :rem 157
310 DATA 12,143,37,162,33,135,14,24,28,49 :rem 182
    ,28,49,18,209,28,49,14,24 :rem 182
320 DATA2,143,42,62,37,162,16,195,33,135 :rem 55
    ,33,135,21,31,33,135,16,195 :rem 55
330 REM SETTA LA VOCE ED IL VALORE DI VEL :rem 167
    OCITA' :rem 167
340 K=54272:P=175:W=K+4 :rem 255
350 POKE54296,15 :rem 99
360 REM SETTA LA MISURA SELEZIONATA DAL V :rem 79
    ALORE DATA :rem 79
370 DATA1,3,6,2,1,4,6,2,3,4,1,5,1,4,6,7, :rem 128
    4,6,2,1,3,6,9 :rem 128
380 DATA 1,1,4,5,1,4,6,2,3,4,1,5,1,4,1,5, :rem 253
    1,4,6,9 :rem 253
390 DATA 1,4,6,2,3,6,1,5,1,4,6,7,3,4,6,2, :rem 138
    1,4,3,7,1,4,6,9 :rem 138
400 DATA 1,4,3,7,1,6,4,5,6,3,6,2,4,6,1,5, :rem 6
    1,4,6,9 :rem 6
410 DATA 1,4,3,7,6,3,6,2,4,6,1,5,1,3,6,7, :rem 235
    3,6,1,5,1,4,6,9,8 :rem 235
420 READRR :rem 86
425 ON RR GOTO520,426,550,560,427,590,428 :rem 15
    ,620,1000 :rem 15
426 Y=12:GOTO650 :rem 163
427 Y=14:GOTO650 :rem 166
428 Y=13:GOTO650 :rem 166
520 Y=1:X=RND(1):IFX<.35THENY=3 :rem 126
530 IFX>.75THENY=2 :rem 78
540 GOTO650 :rem 109
550 Y=10:IFRND(0)<.4THENY=11:GOTO650 :rem 146
560 Y=4:X=RND(1):IFX<.35THENY=5 :rem 135
570 IFX>.75THENY=6 :rem 86
580 GOTO650 :rem 113
590 Y=7:X=RND(1):IFX<.35THENY=8 :rem 144

```

```

600 IFX>.75THENY=9 :rem 83
610 GOTO650 :rem 107
620 PRINT"[CLR]{GIU'}{BLU}BENE,QUESTO E' :rem 126
    TUTTO:SPERO SIA PIACIUTO!" :rem 126
625 POKE53281,1 :rem 44
630 PRINT"[GIU']DAI ANCORA IL RUN PER ASC :rem 136
    OLTARE ALTRI" :rem 136
635 PRINT"CINQUE BRANI!!!":END :rem 20
640 REM LE SEGUENTI SONO LE MISURE USATE :rem 122
    DAL COMPUTER PER CREARE IL BRANO :rem 122
650 POKEW,17:POKEK,L(Y,1):POKEK+1,H(Y,1): :rem 156
    POKEK+7,L(Y,2):POKEK+8,H(Y,2) :rem 156
655 POKEW+7,17:FORQ=1TOP:NEXT:POKEW,16 :rem 187
660 POKEK,L(Y,3):POKEK+1,H(Y,3):FOR=1TOP :rem 115
    :NEXT :rem 115
670 POKEW,17:POKEK,L(Y,4):POKEK+1,H(Y,4): :rem 170
    POKEK+7,L(Y,5):POKEK+8,H(Y,5) :rem 170
675 POKEW+7,17:FORQ=1TOP:NEXT:POKEW,16 :rem 189
680 POKEK,L(Y,6):POKEK+1,H(Y,6):FOR=1TOP :rem 123
    :NEXT :rem 123
690 POKEW,17:POKEK,L(Y,7):POKEK+1,H(Y,7): :rem 184
    POKEK+7,L(Y,8):POKEK+8,H(Y,8) :rem 184
695 POKEW+7,17:FORQ=1TOP:NEXT:POKEW,16 :rem 191
700 POKEK,L(Y,9):POKEK+1,H(Y,9):FOR=1TOP :rem 135
    :NEXT:GOTO370 :rem 135
1000 POKEK,143:POKEK+1,12:POKEK+7,165:POK :rem 206
    EK+8,31:POKEK+14,30:POKEK+15,25 :rem 206
1010 POKEW,17:POKEW+7,17:POKEW+14,17:FOR :rem 63
    =1TO2000:NEXT:POKEW,16:POKEW+7,16 :rem 121
1020 POKEW+14,16:GOTO370 :rem 121

```

A tutti i lettori

Tutti i listati pubblicati sono stati ricavati direttamente dal computer dopo aver eseguito ed accuratamente testato il programma. Non contengono quindi né errori tipografici né errori che comunque impediscano la corretta esecuzione del programma stesso. Vi preghiamo pertanto di prestare la massima attenzione nella copiatura e di non tempestare di telefonate la redazione, in quanto, ripetiamo, tutti i listati pubblicati sono corretti. Per eventuali errori tipografici riscontrati nel testo degli articoli (e quindi fotocomposti), verrà pubblicata una errata corregge sui numeri successivi della rivista.

**Allegre,
Fresche,
Spiritose,
Pratiche.**



**Con tutta la competenza del
GRUPPO EDITORIALE
JACKSON**

Recensioni Software

di M. Cristulib Grizzi

Black Hawk per C64

Thorn Emi sftw.

Vi trovate nei panni di un addestratissimo pilota, imbarcato su una delle più sofisticate e moderne portaerei esistenti, ed il vostro aereo è lo strumento di distruzione più completo finora creato dall'industria bellica: il Black Hawk.



La vostra missione è quella di attaccare e distruggere gli aeroporti e le basi di lancio missilistiche del nemico, evitando e neutralizzando con il vostro aereo la caccia avversaria e lo sbarramento contraereo. Dovrete avere tempi di reazione minimi per evitare elicotteri superarmati, tanks che vi sparano da terra, i colpi della contraerea, missili SAM e razzi sensibili al calore dei vostri jets, che vi inseguono inesorabilmente. Per iniziarvi al combattimento, nella carlinga del vostro aereo è diffusa la "Cavalcata delle Valchirie" di Wagner (le possibilità musicali del C64 non finiranno mai di stupirvi...) e, man ma-

no che procedete nella missione, il computer di bordo calcola la percentuale di obiettivi distrutti, dotandovi di nuove e sempre più sofisticate armi man mano che questa viene migliorata.

È possibile scegliere tra due livelli di gioco, identificati dal nome che assume il vostro aereo: Rookie (principiante) ed Honcho (difficoltà normale); è inoltre possibile fermare il gioco in qualsiasi momento premendo il tasto F3 (F5 per ripartire). Il gioco è composto da due particolari schermate: la prima rappresenta la visione sul monitor del vostro sistema computerizzato d'attacco, degli obiettivi nemici che potete neutralizzare con il sistema elettronico di guida dei missili aria-aria ed aria-terra, mentre l'altra schermata è di carattere difensivo e presenta l'immagine computerizzata del vostro aereo in combattimento con gli elicotteri, i missili ed i tanks dell'avversario. La manovra del Black Hawk è effettuata ovviamente tramite il joystick, ed è estremamente rapida. Con le sue caratteristiche di ottima grafica, di estrema velocità d'azione e di effetti sonori e musicali, Black Hawk è forse uno dei migliori giochi d'azione che siano finora comparsi sul mercato per il C64.

Submarine commander per VIC 20

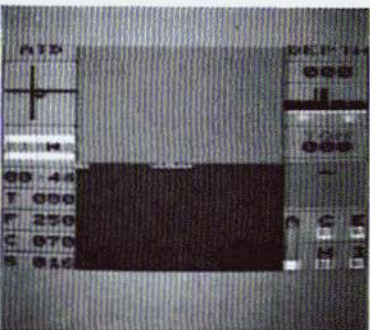
Thorn Emi sftw.

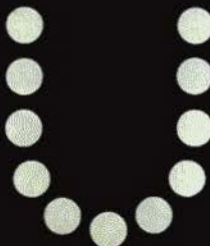
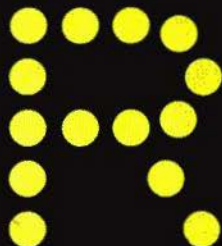
Questa volta vi trovate al comando di un sommergibile in pieno Mediterraneo, alla caccia dei convogli nemici. Sul video appare una mappa del campo di operazioni (in alta risoluzione e realizzata veramente con grande cura), con



un puntino lampeggiante che indica il vostro punto-nave, ed altri puntini in movimento indicanti i convogli nemici in navigazione.

Manovrando timone e potenza dei motori, iniziate a navigare verso l'obiettivo: quando vi trovate in prossimità di quest'ultimo, un segnale d'allarme vi avverte che siete giunti in zona d'ope-





razioni. È quindi opportuno immergersi, tenendo d'occhio i timoni di profondità ed il grafico dell'ecoscandaglio (cozzare, violentemente contro il fondo può costarvi l'integrità dello scafo). A questo punto il sonar vi rivelerà la rotta di avvicinamento; salendo a quota periscopica ed alzando il periscopio potrete finalmente avvistare le sagome delle navi e lanciare i primi siluri. Man mano che colpite i vostri obiettivi, il tonnellaggio affondato viene visualizzato in uno dei tanti display numerici dello schermo. Si gioca con il joystick e la tastiera (per la vastità dei comandi disponibili), ed è veramente appassionante per la completezza delle simulazione e per la complessa articolazione delle possibili situazioni di gioco.

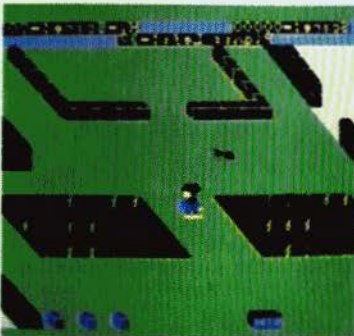
Pedro per C64

Image software

In questo simpaticissimo e frenetico arcade ricoprite il ruolo di Pedro, giardiniere dalle evidenti origini centroamericane, impegnato nella cura della sua modesta aiuola, forse un po' ristretta come dimensioni ma certamente tra le più belle e curate della zona. In basso e sulla destra figurano un mucchio di concime e uno di mattoni, con i quali il nostro sta costruendo un labirinto a scopo ornamentale. Vicino a lui anche

una cassetta di semi: al centro del giardino, i suoi bellissimi fiori, costati mesi di fatiche e cure amorevoli. Ma, improvvisamente, le formiche, poche ma agguerrite, distraggono un fiore dopo l'altro. Egli cerca di contrastarle sbarazzando loro la strada con mucchi di concime e, vista la scarsa resistenza di questi, coi mattoni. Riesce anche a schiacciare sotto i piedi alcuni degli assalitori, ma presto diventa evidente che per respingere la ferocissima orda ci vuole ben altro... e tenta di correre ai ripari precipitandosi verso i semi, per rimpiazzare i fiori distrutti con altri giovani e teneri germogli. Ma ecco un malintenzionato avvicinarsi con aria cupida ai semi! Ancora Pedro deve correre a salvare la sua unica ricchezza, e mettere in fuga il ladro.

Siccome il giardino è sempre stato tutto per lui, è chiaro che la sparizione totale dell'aiuola provocherà la sua morte. Per quanto tempo riuscirà a sopravvivere?



vivere? Le formiche non si vedono più, e c'è il tempo di piantare qualche altro fiore - prima dell'arrivo dei ratti giganti! Passando da un'ondata all'altra, vi ritroverete sempre più coinvolti nel gioco, catturati dalla velocità e dalla precipitazione con cui dovrete condurre la partita. Inizialmente PEDRO sembra un arcade come tanti altri: ma si distacca quasi subito dalla massa, se non altro per la varietà delle situazioni proposte.

PEDRO
(cod. DIGMC01),
SUBMARINE
COMMANDER
(cod. DTEMV01),
BLACK HAWK
(cod. DTEMC01),
sono in vendita

presso:
TECHNOCLUB
via Rosellini, 12
20124 Milano

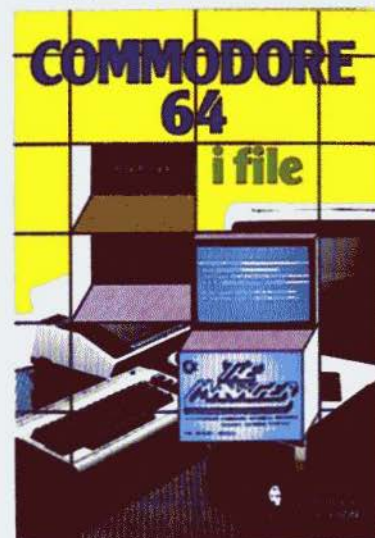
I programmi possono
essere ordinati
usando il coupon
a pag. 82

di M. Cristulib Grizzi

Commodore 64: i file

di Rita Bonelli
ed. Jackson

(Un libro, questo, molto atteso da tanti utenti del C64 che, interessati ad un uso proficuo della macchina, si trovavano tuttavia spaesati di fronte a manuali sibillini ed a libri che solo superficialmente trattavano questo importantissimo argomento. La buona conoscenza delle problematiche di gestione file, infatti, pur essendo fondamentale per sviluppare programmi applicativi sia professionali che "casalinghi", trovava grosse difficoltà ad essere acquisita non tanto per complicazioni intrinseche all'argomento, quanto per mancanza di validi supporti didattici.



"Commodore 64: i file" vuole rivolgersi sia al programmatore esperto che al principiante il quale abbia già acquisito le tecniche della programmazione in BASIC: il libro infatti segue ed approfondisce il precedente volume della

stessa autrice dedicato al BASIC del Commodore 64.

Articolato in tre parti distinte e dedicate rispettivamente ad una trattazione di carattere generale, ai file su cassetta ed a quelli su disco, il libro usa molti esempi in BASIC. Questi ultimi saranno senz'altro utilissimi al lettore non solo per comprendere quanto esposto nella trattazione teorica, ma anche per potersi esercitare in eventuali modifiche ed adattarli alle proprie e particolari esigenze.

L'esposizione dei concetti trattati è scorrevole e piana, ed i listati inseriti nel testo, pur se ottenuti con stampante a matrice di punti, sono tutti ben leggibili e stampati con cura.

In conclusione, questo libro sarà un valido supporto per chi vede nel computer non solo un sofisticato videogioco ma anche uno strumento per l'hobby e per il lavoro.

Easy programming for the Commodore 64 (La programmazione facile per il C64)

di Ian Stewart e Robin Jones
ed. Shiva

In genere, il problema che si presenta ai novizi del Commodore 64 che desiderano imparare a programmarlo, è quello della scarsità di informazioni presenti sul manuale fornito con la macchina.

È disponibile la "Guida di riferimento del programmatore", un supermanuale che comprende anche la descrizione dei pin delle porte di interfaccia, gli intervalli di temporizzazione del microprocessore e tante altre "chicche" per il tecnico, ma non so quante persone tra gli utenti di questo computer possano essere in grado di sfruttarlo appieno. Proprio a costoro si rivolge invece questo libro, che vuole essere una gui-



da attraverso le varie tecniche di programmazione del C64. Lo stile di scrittura è senz'altro molto piano e comprensibile, e non sono certo pochi gli esempi di programmi su tutti i settori nei quali il C64 eccelle, come grafica, musica ed animazioni.

A scopo didattico, vengono presentati alcuni problemini le cui risposte corrette sono elencate alla fine di ogni capitolo: il lettore viene "preso per mano" e, passo dopo passo, viene posto in grado di creare programmi autonomi. Altro punto di forza del libro sono i listati, finalmente ottenuti con una qualità di stampa chiarissima, e senza sequenze interminabili di simboli grafici in "reverse".

Il libro dei giochi del Commodore C64

di Owen Bishop
ed. Jackson

Una collezione di giochi per l'ormai diffusissimo Commodore 64: dal poker agli "arcade", passando attraverso gio-

chi musicali e grafici sia superveloci che "meditativi"....
Ventuno programmi interamente in BASIC e dai listati ben intelligibili, pur essendo stati ottenuti con una stampante VIC 1515. Largo uso viene fatto degli Sprites e delle possibilità grafiche e sonore del C64, ed ogni programma è

ne dell'autore, il quale ha ormai scritto più di trenta libri di programmazione, dedicati ai più diffusi modelli di home computer presenti sul mercato.

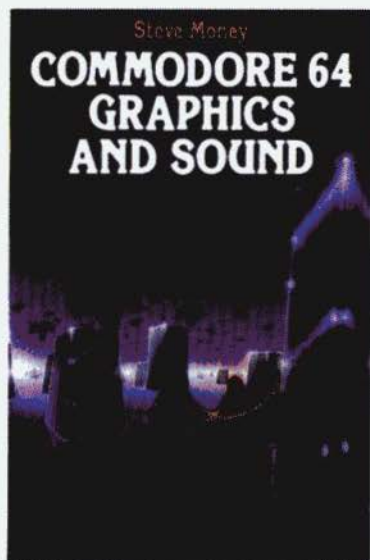
Commodore 64 Graphic and Sound

(Il suono e la grafica del Commodore 64)
di Steve Money
ed. Granada

Una guida pratica per il programmatore interessato alla grafica ed agli effetti sonori, che comprende molti esempi sia in BASIC che in linguaggio macchina. Il libro spiega dettagliatamente le tecniche per tracciare linee nella pagina grafica del C64 al fine di ottenere le forme desiderate, e come costruire dei disegni. Grande risalto viene dato all'uso del colore, degli Sprites e dei caratteri ridefiniti. Una parte è dedicata all'uso delle animazioni - fondamentali per i video games - ed a come ottenere disegni in prospettiva ed effetti tridimensionali. Passando alla parte "sonora" del libro, vengono sviscerati tutti i segreti ed i trucchetti dell'integrato SID, e molti programmi di utilità pratica forniscono un buono spunto di sperimentazione.

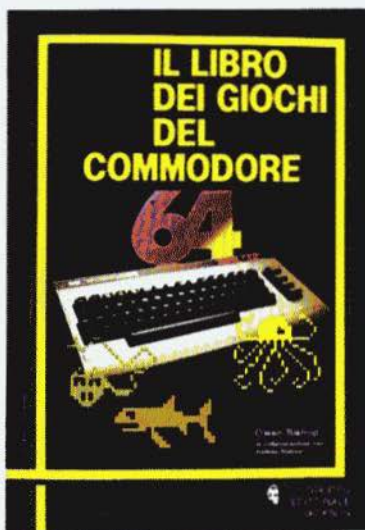
Molto gradite dai programmatori "in proprio" dei giochi saranno sicuramente le molte routines in linguaggio macchina, per realizzare lo scroll fine dello schermo nelle quattro direzioni, per la pulizia della pagina grafica, il plottaggio di punti e linee, etc.

Il libro contiene anche un programma



in grado di trasformare il C64 in una tastiera elettronica, per la gioia di tutti i "musicofili".

**I libri possono
essere ordinati
utilizzando il
coupon a pag. 82**



corredato da complete istruzioni per l'uso dei rispettivi giochi; il modo in cui essi funzionano è dettagliatamente esposto, insieme a suggerimenti sulle migliori tattiche da adottare per ottenere la vittoria.

Vengono anche forniti suggerimenti su come eventualmente modificare i programmi, al fine di adattarli meglio ai propri gusti e preferenze sia grafiche che musicali. Superflua la presentazio-

INFORMATICA
INFORMATICA
INFORMATICA
INFORMATICA



**UNA RIVISTA
DEL GRUPPO EDITORIALE JACKSON**

input.output

Piccoli
annunci

Bellissimi programmi (giochi e utilities) a ottimi prezzi; segnali: grandmaster, calcio, data base su cassetta, speed basic, introd. basic, tennis, grafici ecc... Offro anche gioco "Clowns" + paddles (1 cartuccia). Scrivete o telefonate Carlo Rossi - Via Colombara, 36/5 - 15068 Pozzolo F. (AL) - Tel. 0143/417753

Siete forti (Tanto che mi sono subito abbonato cogliendo al volo la vostra offerta di 15 numeri). Comunque ho un VIC 20 e mi piacerebbe veder pubblicare più utility e meno videogiochi (3,5 KRAM permettendo) Daniele De Donatis - Via San Rocco, 17 - 66010 Villamagna (CH) - Tel. 682565

Vendo C64 con 3 mesi di vita è ancora con garanzia a sole L. 650.000. Cambio C64 con un altro C64 con sistema americano. Motivo: riparto per il Venezuela dove la corrente è di 120 Volt 60 Hz e la TV non usano il sistema Pal affrettatevi, riparto verso metà settembre Pompeyo Silvestri - Via Pavesi, 6 - 67100 Aquila - Tel. 61897

Vendo e cambio software VIC 20 es. (Scrembie Alien Attack ecc). Prezzi modici per informazioni e liste programmi, scrivere al mio indirizzo. Una supersorpresia vi attende per ogni acquisto. Affrettatevi!!! Marcello Mazzolani - Via Sant'erno, 130 - 48020 Sant'erno (Ravenna)

Cause passaggio a sistema C64 vendo videogiochi Atari CX 2600: console + 9 cartucce: tipo pole position, pac man, vanguard, jungle hunt, e altri! Il tutto a L. 450.000. Per informazioni: Davide Bonino - C.so Nizza, 50 - 12100 Cuneo - Tel. 51158

Cambio o vendo per "CBM64" oltre 100 ottimi programmi su cassetta (giochi, utility, ecc.) tutti originali americani ed inglesi in LM. L'annuncio è sempre valido. Inviare o richiedere lista o telefonare ore pasti a: Andrea mamini - Via Cantagalli, 19 - 48018 Faenza (RA) - Tel. 0546/25474

Vendo per VIC 20: Motherboard 3 slot, che consente di collegare fino a 3 cartucce contemporaneamente: L. 34.000. Supermotherboard 6 slot, con interruttori e reset L. 50.000 light pen (anche C64) L. 34.000, espansione 3K autocostituita, perfetta L. 38.000 Gianni Bozzi - Via Savona, 16/5 - 20099 Sesto San Giovanni (MI) - Tel. 02/2407825

Vendo e cambio programmi per VIC 20 (oltre 100, quasi tutti in LM), prezzi modici. Mi interessano soprattutto utilities, programmi in LM e compilatori (max 3K di espansione). Inoltre scambierei cartuccia super expander con espansione da 8K. Carrone Vincenzo - Via G. Pascoli 67 - 86100 Campobasso tel. 0874/91995

Un consiglio: continuate a pubblicare i programmi per VIC 20 anche in versione per C64 Andrea Cutrona - Via Ofelia, 35 - 95125 Catania - tel. 352251

Innanzitutto complimenti per la rivista (attesa da tanto). Nella rivista speravo di trovare dei programmi per VIC 20 (ver. base) che facessero dei grafici o figure sullo schermo (es. cuori, cerchi, ecc.) Giuseppe Buttici - Via Belvedere, 41 - 92010 Giardinia Gialliotti (AG)

Per CB. 64 vendo software di tutti i generi in linguaggio macchina. Il tuo computer ha bisogno di... mangiarli!!! Richiedete lista telefonando a: Simone Borgheresi - Via Montesanto, 17 - 59036 Poggibonsi (SI) - tel. 937336

Vendo e cambio per Commodore 64 oltre 200 giochi a prezzi convenienti tutti i migliori, molti "adventure" possibilità di sconto se comprate grandi quantità di software, utility. Molto disposto al cambio. Stefano Di Paola - Via Tazio Nuvoletti, 205 - 00142 Roma - Tel. 5036108

Il credo che si potrebbe svolgere in questo modo: voi decidete il programma e i partecipanti invieranno i loro record, i loro dati e una foto in cui si trova il record Emanuele Borlini - Via Marabina, 95 - 48100 Ravenna - Tel. 0544/433032

Desidero se è possibile informazioni sull'impiego della tavoletta grafica, quali sono le marche che producono queste e quali sono le più economiche. Albino Montella - Via Verona, 13 - 84091 Battipaglia (SA) - tel. 0828/72107

Vendo VIC 20 + registratore C2N-B + cartuccia super exp. + tutto software anche in LM. Tutto a L. 120.000 trattabili telefonare ore pasti o scrivere a: Antonio Babini - Via Regnoli, 58 - 40138 Bologna - Tel. 260279

Vendo VIC 20, poche ore di funzionamento per L. 160.000. Microfono preamplificato da tavolo in ottimo stato L. 60.000. Telefonare dopo le ore 15e fino alle 22.00 Gerardo Izzo - Via Bellini, 1 - 81042 Calvi Risorta (CE) - Tel. 0823/873026

Vendo C64 mai usato, garanzia in bianco + 15 giochi a L. 700.000 (compreso il registratore). Vendo molti programmi per C64, qualche esempio: Dr Jand, Larry Bird go one on one, flight simulator II, ecc. Elenco gratis a richiesta. Stefano Calcaterra - Via Marconi, 34/2 - 40127 Bologna - Tel. 051/521063

Vendo programmi su cassetta per Computer Commodore VIC 20. Cerco Computer ZX80 oppure ZX81. Per informazioni scrivere o telefonare a: Mario Boemi - Via Gerubino Pilli Pal. T. Sc. 13 - 98010 Messina - Tel. 090/772643

Vendo e cambio il cartridge "Cosmic cruncher" nuovo e compro o cambio programmi (soprattutto giochi) per VIC 20 Daniele D'Andrea - Via del Larici, 22 - 00171 Roma - Tel. 253851

Siamo due amici ed abbiamo fondato il Vic Club. È prevista la spedizione mensile di software ricavata da riviste specializzate. Per iscrizioni inviare L. 5.000 mensili a mezzo vaglia. Per eventuali informazioni scrivere a: Mariano Marsicano - Via Panoramica, 85 - 80056 Ercolano (NA) tel. 081/7396011

Complimenti per l'ottima rivista e per l'attentissima gamma di software: vorrei però suggerire anzi richiedere, se possibile, la pubblicazione di qualche programma di utility o magari per la grafica in alta risoluzione. Angelo Ariando - Piazza degli Angeli, 2 - 20146 Milano - tel. 465780

Utenti di CBM 64! Scrivetemi e formeremo un club ed una ricca biblioteca di programmi cui tutti potranno accedere!!! Massimo Orsi - Via G. Verdi, 12 - 22046 Manguzzo (CO) - Tel. 031/650085

64 Commodore Club, scambio, vende, compra software per 64; grazie a parenti americani, disponiamo dei migliori programmi applicativi e ricreativi con continue novità USA. Per informazioni e cataloghi scrivere a: Giuseppe Sciotti - Via Vignizolo, 23 - 25018 Montichiari (BS)

Per C64 cambio, compro e vendo software di qualsiasi genere, preferibilmente su disco! Dispongo di 380 titoli. Cambio anche con stampante se in buone condizioni. Massimo Pinciroli - Via Thiene, 7 - 21052 - Busto Arsizio - tel. 0331/684348

Vendo per CBM64 cassetta contenente i seguenti giochi: donkey kong, Q bert, frogger, centipede, arcedia, super pipeline, tutti registrati con il turbo tape 64 (compreso in cassetta) che ha la capacità di aumentare la velocità del registratore di 16 volte. Il tutto a L. 25.000 (comprese spese postali) Salvo D'Urso - Via La farina, 3d - 95018 Riposto (CT) - tel. 095/936075 (ore 13-17)

Compro per CBM 64, programmi musicali su cassette, e di utility. Gradito anche scambio opinioni su esperienze musicali sul CBM 64. Giorgio Rizza - Via Raccomandata, 24 - 97015 Modica (RG) - tel. 0932/944434

Compro, vendo e cambio programmi per VIC 20. Mi interessano soprattutto programmi di giochi in LM espansi a 8/16K e quelli tratti da cartuccia. Scrivetemi ed eventualmente inviatemi la vostra lista. Gaetano Stornante - Via Episcopo, 84 - 96011 Augusta (Siracusa) - Tel. 0931/974902 (dalle ore 14.00 alle 15.30)

Vuoi comprare software per il tuo commodore 64 su cassetta o disco? Richiedimi il catalogo dei mie 900 programmi moltissime novità in anteprima, massima serietà. Luigi beviglia - Casella Postale 41 - 21052 Busto Arsizio (VA)

Vendo per VIC 20 i migliori programmi oggi in circolazione, invio catalogo dettagliato (36 pagine). Cedo anche traduzione istruzioni super espansione, tool kit, mon. Luigi De Negri - Via Puggia, 22 - 16131 Genova (GE)

Vendo due cassette della rabbit per VIC inespanso: pakarama e skramble nuove appena comprate solo a milano e a L. 10.000 trattabili. Mirko Bertoni - Via Le Monzie 363 - 20128 Milano - Tel. 2573616

input-output

Piccoli
annunci

Vendo C64 + registratore 1530 + unità floppy 1541 + "easy script" + "80 colonne" + "Screen Graphics" + molti programmi: Grafica, statica utilità (tutti i manuali in italiano) valore del blocco circa L. 1.900.000 cede tutto a L. 1.300.000 (causa doppio regalo) telefonare ore pasti a: Fabio Fraternali - via Venti, 10 - 61029 Urbino (PS) - Tel. 0722/37576

Per Commodore 64 vendo cambio tantissimi programmi di qualsiasi genere (giochi in L/M utility, gestionali, linguaggi) su disco o nastro a prezzi eccezionali. Scrivere per invio dettagliato listino. Telefonare per scambio. Stefano Dominioni - Via Tommaso, 16 - 21100 Varese - Tel. 0332/229909

Compro cambio e vendo a poco prezzo software su cassetta per VIC 20 inesperto, vendo anche un cartidge: road race. Chiedetemi la lista e ve la manderò. Telefonare ore serali ed ore pasti a: Giancarlo D'Angelo - via A. Pirelli, 12 - 00146 Roma - Tel. 06/5581008

Per Commodore 64 vendo tantissimi programmi (giochi in L/M, utility, gestionali, linguaggi) a prezzi eccezionali, ad esempio Buck Rogers L. 8.000, Turbo tape, Simon, forth L. 10.000, Archivio L. 8.000, Koaia L. 10.000. Scrivere per invio listino. Telefonare per scambi. Stefano Dominioni - via Tommaso, 16 - 21100 Varese - Tel. 0332/229909

Vendo cambio per CBM 64 programmi su cassetta dei giochi formidabili come quelli da bar. Per informazioni scrivere a: Andrea Pistolesi - via Palermo, 2 - 56020 Cadanne (Pisa)

Cambio software per C64. Inviare la lista e io invierò la mia. Alessandro Flammini - via F. Ruliano, 19 - 00175 Roma

Interfaccia per qualsiasi registratore per C64 e VIC 20 vendo a L. 35.000 contrassegno - s.p. con garanzia max celerità Romano Paolini - via F.M. Briganti, 397 - 80144 Napoli

Compro, vendo, cambio software per Commodore VIC 20. Gilberto Ferrari - Bolleri, 314 - 38040 Martignano (TN)

Vorrei sapere qualcosa in più a proposito della grafica ad alta risoluzione, cioè come ottenerla come sfruttarla. vorrei inoltre sapere se esistono valide pubblicazioni (per principianti) inerenti il LM del 64 Luca Prandini - Corso Italia, 10 - 58014 Orbetello (GR) - tel. 0564/867147

Vendo o scambio programmi per C64. Arcadia, le mans, pole position, frogger, soccer, sky-race, simon's basic, easy script, magic desk per speed... cassette vergini C10 L. 1.000 C20 L. 1.300. Qualsiasi quantità. Ennio Parmigliani Via E. Porro, 47 - 43039 Salsomaggiore T. - tel. 0524/76327

Perché non dedicate un paio di pagine al linguaggio macchina del VIC 20 e del C64? Sarebbe certamente molto utile! Potreste pubblicare un programma che faccia sapere la memoria occupata da un programma? (per il VIC 20) Mauro Godi - Via Strà, 5 - 37042 Caldiero (VR) - tel. 7650315

Vendo programmi per VIC 20 (giochi ed altri) su cassetta. Ne possiedo in Basic e in LM. I prezzi sono irrorati e i giochi specialmente sono fantastici. Possiedo anche programmi per fotocollante e bilancio familiare. Scrivere o telefonare. Annuncio sempre valido. Alberto Ferrara - Via dell'Industria P.zza Iuvare - 87012 Castrovillari - tel. 0981/22845-931041

Vendo o cambio ottimi programmi LM e Basic per VIC 20. Ne possiedo circa 200. Prezzi incredibili. Totocalcio, Blitz, Galaxia, ecc. Richiedere il catalogo con la denuncia di ogni prog. L. 800 Giuseppe De Frenza - Via M. Raineri, 58 - 70028 Sannicandro di Bari

Vendomi programmi di vario genere, assolutamente originali e a prezzi concorrenziali tra cui: Simons's basic, snapman, tennis 3D, programmi grafici, musicali, dietetici, gestionali, ecc. Tool kit, cbm mon, frogger, crazy climber, bioritmi. Si effettuano, inoltre, lavori software su commissione. P.zza Garibaldi, 37 - 70122 Bari - tel. 080/232142

Cartidge "mare master" della hewlett. Nuova prezzo interessante. Telefonare ore pasti. Giov. battista Fornari - Via Villorosi, 8 - 20099 - Sesto San Giovanni (MI) - tel. 2474960

Vendo programmi (giochi) per VIC 20. Prezzi bassissimi. Scrivere per richiesta di catalogo giochi con prezzi. Massima serietà. Massimo Tucci - Via A. Manzoni, 157 - 80123 Napoli - Tel. 640063

Vendo per VIC 20 e CBM 64 interfaccia per poter usare qualsiasi registratore a sole lire 25.000. Telefonare ore pasti a: Daniele De Donatis - via S. Rocco, 17 - 66010 Villamagna (CH) - Tel. 682565

Vendo al prezzo di L. 50.000 15 giochi per Commodore 64 (Suicide strike, One on one, Pooyan, Ladi tut, Calcio, Pole position, Hunc, Frogger, dama, Motor mania, Fort apocalyptor, crazy kong, torre di anoy, per man, griddier). Valerio Manca - viale dei 4 venti, 104 - 00152 Roma - Tel. 06/5813562

Vendo 2 cartidges per VIC 20 - "Mission impossible adventure" e "road race" - in ottimo stato (usati due settimane), al prezzo trattabile di L. 25.000. Fernando Mastidoro - via Campania, 99 - 74100 Taranto - Tel. 099/373046

Per CBM 64 scambio o vendo programmi di ogni genere, per la vendita i prezzi sono bassissimi. L'annuncio è sempre valido. Enzo Petrizzo - via Nazionale - 84030 Padula Scalo (SA) - Tel. 0975/74022

Vendo cassetta per Commodore 64 con 16 programmi per lire 50.000. I programmi contengono istruzioni in italiano. Scrivere a: Luca Angiolini - via A. Vivaldi, 10 - 52100 Arezzo - Tel. 075/380774

Vendo, cambio programmi per CBM 64 in fotocopia o cassetta anche il linguaggio macchina, massima serietà. Gaetano D'Agostino - via Ofelia, 35 - 95124 Catania - Tel. 358642

Cambio programmi per CBM 64 e vendo cassetta di manic miner per Commodore 64 superba grafica in alta risoluzione e ottimo suono al prezzo di L. 15.000, telefonare ore pasti o scrivere a: Massimo Bianco - c.so Corsica, 137 - Torino - Tel. 618930

Vendo e cambio per VIC 20 software: listati o nastri con giochi per versione basic, 3K, 8K, 16K. Inviare L. 1000 per lista (anche in francobollo) prezzi: Listati L. 1.500. Giochi da L. 3.000 - 5.000 - 9.000 compreso nastro. Escluso spese postali. G. Carlo Peretti - via Odeschi da Gubbio, 18 - 00146 Roma - Tel. 06/5580689

Cambio software per VIC 20 giochi utility. Escludo commercio. Aiuto principianti solo contro rimborso spese. Rispondo a tutti. Annuncio sempre valido. Alvaro Ceccarini - via Di Vittorio, 10 - 58022 Follonica (GR) - Tel. 0566/43248

Vendo Videopac Philips + 2 cassette, al prezzo di L. 300.000 Fabio Cacciola - via Sorrento trav. 8* n. 45 - 89100 Reggio Calabria - Tel. 0965/55787 oppure 54920

Compro, cambio, vendo programmi per VIC 20. Invoio una cassetta registrata a chi ne invia una. Compro programmi tratti da cartuccia e espansi 8/16K specialmente in L/M Gaetano Stornante - via Epicarmo, 84 - 96011 Augusta (Siracusa) - Tel. 0931/874902 - dalle 14 alle 15.30

Per Commodore 64 vendo e scambio programmi su cassetta e cartuccia. Turbo tape, Simon basic, Frogger, Pacman, Calcio, ecc. Per chi acquista 5 giochi, in regalo 2. Telefonare o scrivere a: Fabio Manzo - via Calabria, 36 - 81022 Casagiove (Caserta) - Tel. 0823/466177

Per C 64 vorrei acquistare dei programmi gioco con supporto cassetta o listato. Per vendite, inviate lista, caratteristiche, prezzi a: Cesare Marcolini - via dei Velini, 65 - 62100 Macerata - Tel. 0733/31754

Causa passaggio a sistema superiore vendo VIC 20 + registratore + software in basic e LM + listati + 4 manuali + 30 joystick + imbollo originale, il tutto in perfette condizioni a L. 400.000 (trattabili) Massimiliano Marcellino - V.le Ofanto, 301 - 71100 Foggia - Tel. 0881/33401

Vendo, cambio listati di giochi e utility su qualsiasi computer, (C64 in particolare). Inoltre lista di giochi per cambio con la mia lista. Grazie Arturo Assennato - Via Archimede, 2 - 94100 Enna - tel. 22219

SUPERVIC

CEDOLA DI ORDINAZIONE - LIBRI
da compilare e spedire in busta chiusa a
J.soft - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

Ordino i seguenti libri per un importo totale di L. + L. 2.000
come contributo fisso per spese di spedizione

Cod. Cod. Cod.

Cod. Cod. Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp. n° 19445204 intestato a J.soft - Milano

Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

LIBRI

COMMODORE 64 - I FILE
IL LIBRO DEI GIOCHI DEL
COMMODORE 64
EASY PROGRAMMING FOR THE
COMMODORE 64
COMMODORE 64 - GRAPHICS
AND SOUND

cod. AJAC037 L. 15.300

cod. AJAC034 L. 11.700

cod. BSHT004 L. 17.300

cod. BGRC008 L. 17.300

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:

Cod. F. e P. Iva

Data

Firma

Per i soci residenti all'estero — pagamento anticipato (vaglia o versamento su n. ccp)

SUPERVIC

CEDOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.soft - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

Ordino le seguenti cassette per un importo totale di L. + L. 2.000
come contributo fisso per spese di spedizione

Cod. Cod. Cod.

Cod. Cod. Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp. n° 19445204 intestato a J.soft - Milano

Pagherò in contrassegno al postino al ricevimento delle cassette

SOFTWARE

PEDRO PER COMMODORE 64
SUBMARINE COMMANDER PER VIC 20
BLACK HAWK PER COMMODORE 64

cod. DIGMC01 L. 14.000

cod. DTEMV01 L. 16.000

cod. DTEMC01 L. 18.000

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:

Cod. F. e P. Iva

Data

Firma

Per i soci residenti all'estero — pagamento anticipato (vaglia o versamento su n. ccp)

SUPERVIC è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

.....
.....
.....
.....
.....
.....
.....
.....

Nome Cognome

Via C.A.P.

Città Tel.

**SUPERVIC
INPUT/OUTPUT**

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o scambiare hardware o software può inviare il tagliando a J.soft - Via Rosellini, 12 - 20124 MILANO

COMPRO VENDO CAMBIO
ZX80 ZX81 ZX Spectrum PERIF. SOFTWARE

.....
.....
.....
.....
.....
.....
.....
.....

Nome Cognome

Via C.A.P.

Città Tel.

OGNI VENERDI' IN EDICOLA

PAPER
soft

IL PRIMO SETTIMANALE DI SOFTWARE SU CARTA
PER IL TUO PERSONAL COMPUTER
Una pubblicazione della J. soft editrice



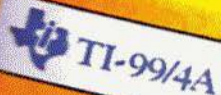
PAPER

L. 1.000

Anno I - N° 8 - 3 agosto 1984



Editor di schermo
Galaxia



Dedale 3-D
Puzzle



Difesa
Meteore
Carica DATA



Nevets
Poker



Nevets
Intre



PAPERSOFT,
il primo settimanale di software
per il tuo home e
personal computer
a sole 1.000 Lire.

J. soft editrice - via Rosellini, 12 - 20124 Milano
tel. 02/6888228-683797

PERSONAL COMPUTER: le prime, le migliori!

PERSONAL SOFTWARE

L'unica che presenta software
per tutti i personal:

Commodore, Apple, Sinclair, T.I., HP,
Sharp, Sega, Olivetti, ecc.



BIT: La più letta, la prima e più diffusa
TEST: Decision Mate V
SUPERBIT - 64 pagine di programmi
per i vostri personal computer.



Con tutta la competenza del
**GRUPPO EDITORIALE
JACKSON**